Shape control of deformable objects by adaptive visual servoing

Romain Lagneau

THESE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES RENNES

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *informatique*

Par

Romain LAGNEAU

Shape control of deformable objects by adaptive visual servoing

Thèse présentée et soutenue à Rennes, le 04/12/2020
Unité de recherche : IRISA
Thèse N° : 20ISAR 24 / D20 - 24

Rapporteurs avant soutenance :

Andrea CHERUBINI	Maître de conférences, Univ. Montpellier
Christian DURIEZ	Directeur de recherches, INRIA Lille

Composition du Jury :

Président :	François CHAUMETTE	Directeur de recherches, INRIA Rennes
Examinateurs :	Tamim ASFOUR	Professor Karlsruhe Institute of Technology
	Andrea CHERUBINI	Maître de conférences, Univ. Montpellier
	Christian DURIEZ	Directeur de recherches, INRIA Lille
	David NAVARRO-ALARCON	Assistant Professor, Hong Kong Polytechnic University
Dir. de thèse :	Maud MARCHAL	Maître de conférences, INSA Rennes
Co-dir. de thèse :	Alexandre KRUPA	Chargé de recherches, INRIA Rennes

Acknowledgement

I would like to thank the dear members of the jury for having accepted to review my work. It was a honor for me to be evaluated by reknown researchers. I would also like to express my gratefulness to my coworkers of IRISA from the Hybrid and Rainbow teams to be such wonderful people. I would also like to thank Fabien and Thierry for their help during these three years, as well as the kind people working in the IT department. I will now switch to French.

Je voudrais tout d'abord remercier ma famille, qui m'a soutenu tout au long de ce projet par leur amour et leurs encouragements. Vous êtes des personnes formidables et je ne saurai jamais assez vous exprimer ma reconnaissance et mon amour.

Je voudrais ensuite remercier mes amis d'enfance: Camille, Brendan, Amaury, Jérémy, Océane; pour leur soutien indefectible tout au long de ces années et leurs qualités trop nombreuses pour être énumérés ici. Je tiens à vous comme je tiendrais à des frères et soeur.

Je voudrais aussi remercier mes amis de jeu de rôle: Amaury, Clément, Gabriel, Julien, Morgane (et par anticipation Benoît); pour ces parties aussi riches en rires qu'en aventure. J'attends toujours la prochaine séance avec impatience pour vous retrouver.

Je voudrais aussi remercier des amis qui, bien que rencontrés plus récemment, comptent énormément pour moi: Clément, Gwen, Audrey, Benoît, Caroline, Thierry, Gabriel. Les moments passés ensemble sont toujours plein de rires et de gentillesse, et je ne vous remercierai jamais assez pour cela.

Enfin, je voudrais remercier une personne qui, bien que rencontrée que très récemment, a pris une place importante dans ma vie. Elle a su m'encourager et m'aider dans les moments difficiles et je lui serai toujours reconnaissante pour cela. J'espère pouvoir me montrer à la hauteur et la rendre aussi heureuse qu'elle ne me rend heureux.

Table of Contents

NOTATIONS

The notations used in this manuscript are the following:

- scalars are represented by normal letters, such as n
- vectors are represented by bold lowercase letters, such as $\mathbf{v}$. The notation v_i denotes its i^{th} component,
- the scalar product between two vectors is denoted by $\mathbf{a} \cdot \mathbf{b}$, while the cross-product between two vectors is denoted by $\mathbf{a} \wedge \mathbf{b}$,
- the Euclidean norm of a vector is denoted by $\|\mathbf{v}\|$,
- matrices are represented by bold uppercase letters, such as $\mathbf{M}$. The value located at the i^{th} row and j^{th} column is denoted M_{ij},
- $\hat{\mathbf{v}}$ represents an estimate of the vector $\mathbf{v}$ while $\mathbf{v}^*$ represents its desired value,
- tensors are represented by overlined uppercase letters, such as $\overline{\mathbf{T}}$. The value located at the i^{th} row and j^{th} column is denoted $\overline{\mathrm{T}}_{ij}$.

In this manuscript, except when otherwise specified, ${}^B\mathbf{x}_A \in \mathbb{R}^3$ corresponds to the vector representing the 3D position of A with regard to the Cartesian frame B. The notation $\mathbf{x}_A$ is an abbreviation for the vector representing the 3D position of A with regard to the world Cartesian frame.

Moreover, ${}^B\mathbf{v}_A = ({}^B\boldsymbol{v}_A, {}^B\boldsymbol{\omega}_A) \in \mathbb{R}^6$ corresponds to the velocity screw vector expressing the velocity of A with regard to the Cartesian frame B, where ${}^B\boldsymbol{v}_A = (v_x, v_y, v_z)^T$ corresponds to the 3D linear velocity vector while ${}^B\boldsymbol{\omega}_A = (\omega_x, \omega_y, \omega_z)^T$ corresponds to the 3D angular velocity vector. The notation $\mathbf{v}_A$ is an abbreviation for the velocity screw vector expressing the velocity of A with regard to the world Cartesian frame.

Finally, k will denote the discrete time. Be t the continuous time, t_0 the initial time and Δt the sampling time, the notation k is an abbreviation for $t_0 + k\Delta t$.

Chapter 1

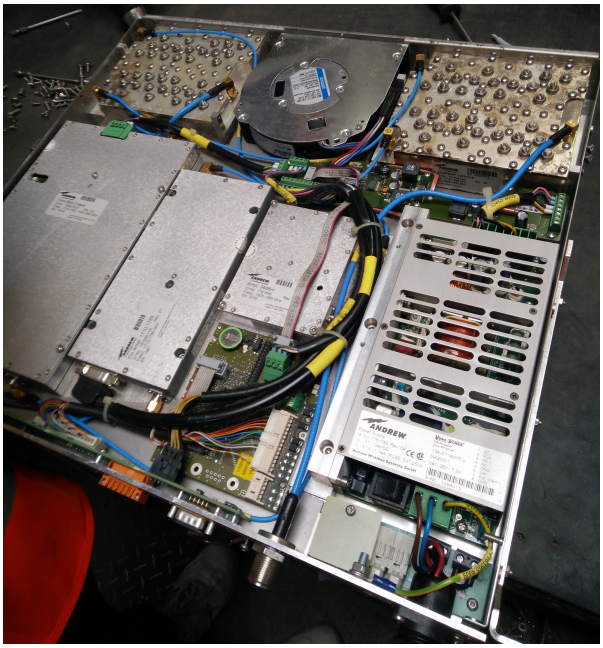
Introduction

Contents

1.1 Context

The work accomplished in this thesis was funded by the *Imagine project* (URL), which is supported by the European Union's Horizon 2020 research and innovation program. The Imagine project involves several European universities and an industrial company Electrocycling (URL). This project aims to address some challenges arising from the automatic recycling of electronic devices, that might include many deformable parts such as wires, thanks to robots. The recycling of electronic devices is of growing interest as the number of existing electronic devices has drastically increased over the past decades. The construction of these devices require some materials existing in limited quantities. Moreover, electronic components tend to be harmful on the environment if they are released in an uncontrolled manner. As of today, the complexity of these devices makes that most of the recycling tasks are performed by human operators. An example of electronic device recycling task performed by a human operator is shown in Figure 1.1. As shown in Figure 1.1a, the electronic device contains a lot of wires that prevent the removal of some of its internal components. A human operator must manually cut these

wires to permit their recycling and to continue recycling of the other components, as shown in Figure 1.1b. Risks to the operator's health, due to the presence of some hazardous chemical products, and human resource cost of this task illustrate why not all electronic devices are recycled. Enabling the automatic recycling of electronic devices using robots is a challenging yet beneficial field of research.

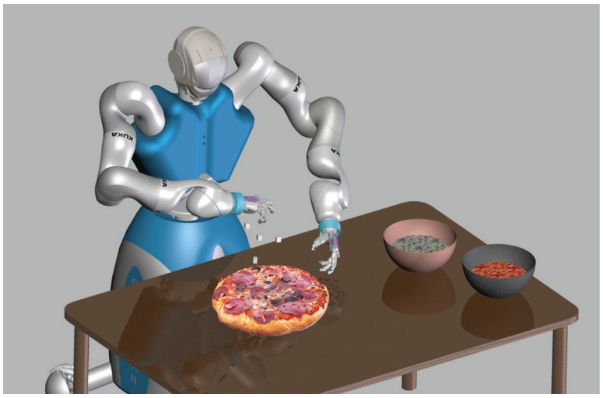


(a) Overview of the interior of an electronic device being recycled.

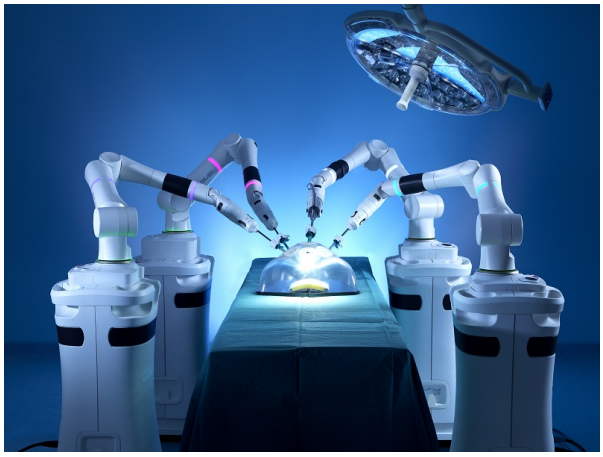
(b) Human operator cutting the wires manually for recycling the electronic device.

Figure 1.1: Pictures presenting the recycling of an electronic device. One can notice the deformable wires that need to be manipulated.

However, the dexterous robotic manipulation of non-rigid objects is more difficult than the one of rigid objects, due to the deformations that occur when forces are applied. These deformations include, but are not limited to, stretching, torsion or compression . Food products are good examples of non-rigid objects with a wide variety of shapes and deformation behaviors. For instance, a pizza dough undergoes both elastic and plastic deformations when it is interacted with. Consequently, it requires specific methods to permit a robot to manipulate it as depicted in Figure 1.2a. Similarly, a tomato slice that needs to be placed on the pizza will require a different amount of force to be grasped without being damaged than a slice of pepper. From Figure 1.2b, one can infer that depending on the number of contact points and their location on the phantom, the latter will deform differently. Moreover, a real body would not deform the same way at different locations due to its material heterogeneity.



(a) Humanoid robot baking a pizza. Figure from the RoDyMan project



(b) Anthropomorphic robot arms working on an abdominal phantom. Figure from the VERDICT Medical Devices website

Figure 1.2: Illustrations depicting possible use-cases for robotic interactions with deformable objects.

The various deformation behaviors of non-rigid objects make it difficult for robots to manipulate them. Specific methods are required to take into account the deformations that arise when external forces are applied to them. Consequently, this work focuses on proposing novel methods for the automatic manipulation of deformable objects. This would allow, for instance, efficient recycling of electronic components or manipulating non-rigid food products. The automatic control of the deformations that a non-rigid object undergoes during its manipulation by one or several end-effector(s) is called *deformation servoing*. In our work, we decided to propose deformation servoing methods relying only on visual information, since visual sensors are widely used in industry, cheap and non-invasive. Moreover, in an industrial context, manipulated deformable objects are usually not perfectly known and observed using non-invasive sensors such as a remote visual sensor.

1.2 Scientific challenges

Different challenges arise from the dexterous robotic manipulation of non-rigid objects.

Challenge #1 Permitting robots to manipulate non-rigid objects while controlling their shape:

Control can either refer to preserving the rest shape, which is particularly important for tasks such as pick-and-place, or to reach a desired deformed shape, which is of interest for

instance in food manufacturing tasks. Several difficulties arise from the robotic manipulation of non-rigid objects. First, it is necessary to be able to track in real-time the deformations that the object undergoes while being manipulated. Then, a relation must be established between the robot motions and the resulting deformations. Finally, the method used to control the robot must be adaptive so as not to build an entirely new system when facing a new object and/or situation.

Challenge #2 Predicting the deformation behavior of a non-rigid object:

As mentioned previously, non-rigid objects of different nature but subject to the same perturbations will not deform the same way. One possibility to permit robotic manipulation of non-rigid objects consists in using methods based on deformation models of the objects and/or their interactions with their environment. Methods based on such models are called *model-based methods*. Setting up of a deformation model requires to acquire a great amount of data to determine physical laws and the geometry of the object that best match the observations. This kind of methods usually allows the manipulation of non-rigid objects with great accuracy. *Model-free methods* attempt to give to robots the ability to understand the relation between their actions and the deformation that a non-rigid object undergoes without relying on a physics-based model. To do so, it is necessary to have deformation tracking capabilities to observe the evolution of the object shape over time. The evolution of the object shape must be related to the robot motions in order to be able to determine the robot actions leading to the desired deformation. This relation must be estimated online as the object deformation behavior may not be consistent over time. To permit dexterous manipulation, this must be performed in the smallest amount of time possible. However, since these methods do not use a physics-based model of the object, they can be very sensitive to measurement noise.

Challenge #3 Determining the deformation model to use while avoiding damaging the object of interest

To avoid damaging the non-rigid object, the determination of the deformation model and the physical parameters describing the object is usually performed before the manipulation. It is particularly important when the non-rigid object of interest is an organ that must be manipulated during surgery. The complexity of the deformation behavior can dramatically increase if the object is non-homogeneous, which is often the case for instance in living tissues. The estimation of the deformation behavior of an object requires an accurate tracking but must not come at the cost of the time performance. Developing novel non-invasive estimation methods that permit to estimate complex behaviors remains a challenge and an active field of research.

1.3 Thesis objectives

This thesis aims to address several of the aforementioned challenges.

1.3.1 Controlling the deformations

First, this thesis addresses the challenges **#1** and **#2** by aiming to control the deformations of various kinds of non-rigid objects without requiring heavy preliminary steps. We chose to develop a model-free method to meet these two requirements. Indeed, as model-free methods are not relying on a model, they can be used on a greater variety of objects . We propose to take inspiration from visual servoing to derive a method that permits to automatically control the deformations of a non-rigid object produced by one or several robotic manipulator(s). This choice was motivated by the fact that visual sensors are widely used in industry, cheap and non-invasive. An adaptive scheme permits to adjust the control law over time according to the observed effects of the robotic manipulator(s) motions on the object deformation. The robustness against noise of the proposed method is ensured by a confidence criterion used during the adaptive scheme.

1.3.2 Manipulating linear deformable objects using robots

The second objective of this thesis is to address the challenges **#1** and **#2** by aiming to dexterously manipulate linear deformable objects using robots in real-time. In order to generate complex shapes, it is necessary to hold both ends of the wire, thus requiring bi-manual robotic manipulation. This leads to new challenges due to the intertwined influence of the extremities motions on the deformation the wire undergoes. To address these challenges, we propose a geometric model of the linear deformable object to derive a new kind of visual feature that can be estimated online. We decided to use 3D B-splines, as they are known to efficiently represent complex linear 3D shapes. Along with a set of visual features we propose an update scheme for the geometric model to adapt accordingly to the visual information captured by the sensor. The proposed visual feature set is considered as the input of a model-free deformation servoing method to control the deformation of different types of wires using two robotic end-effectors.

1.3.3 Estimating elasticity parameters interactively

Sometimes, model-free methods do not provide an accuracy meeting the requirements of the task at hand. In these cases, the use of model-based methods is mandatory. However, this requires knowing the physical parameters of the object that needs to be manipulated. This thesis thus addresses the challenges **#2** and **#3** by aiming to estimate some elasticity parameters of unknown deformable objects interactively. The estimation is performed while the unknown deformable object is deformed by a robotic end-effector. The parameter elasticity estimation uses the measurements obtained from a force sensor and the results of the deformation tracking. The results of the parameter elasticity estimation are used to update the deformation tracking parameters, thereby closing the loop. Once the elasticity parameter estimation has converged, it is possible to remotely estimate deformation contact forces applied to the object using only visual information.

1.4 Contributions and organization of the manuscript

This manuscript is organized as follows. First, Chapter 2 presents a literature review of some methods of the state-of-the-art addressing either of the challenges arising from the robotic manipulation of deformable objects. Then, Chapter 3 presents a novel model-free deformation servoing method, which adapts the motions of one or more end-effector(s) to the resulting deformations undergone by the non-rigid object being manipulated, to generate a desired deformation. The accuracy and stability of the method were compared to state-of-the-art methods. Then, Chapter 4 presents an adaptation of this method to handle dual-arm robotic manipulation of deformable wires. To this end, a novel set of visual features has been proposed in order to be able to estimate online the shape of a deformable wire. Finally, Chapter 5 presents a method that permits to simultaneously track a deformable object and to estimate some of its elasticity parameters. A use-case of the elasticity parameters to perform remote force estimation by vision is also presented. Finally, Chapter 6 concludes this manuscript by presenting short-term and long-term perspectives related to the work presented here.

1.5 Publications from the author

The contributions presented in this manuscript resulted in several publications.

International journals

- R. Lagneau, A. Krupa and M. Marchal, *Automatic Shape Control of Deformable Wires based on Model-Free Visual Servoing*. IEEE Robotics and Automation Letters, volume 5, issue 4, pp. 5252–5259. 2020. Video: https://youtu.be/T9wDX4mdvSw

International conferences

- R. Lagneau, A. Krupa and M. Marchal, *Active Deformation through Visual Servoing of Soft Objects*. Proceedings of the IEEE International Conference on Robotics and Automation. pp. 8978-8984. 2020. Video: https://youtu.be/iovQUcyyuwc

- A. Sengupta*, R. Lagneau*, A. Krupa, E. Marchand, and M. Marchal. *Simultaneous Tracking and Elasticity Parameter Estimation of Deformable Objects*. Proceedings of the IEEE International Conference on Robotics and Automation. pp. 10038-10044. 2020. (* both authors contributed equally to this work.). Video: https://youtu.be/3Ht-xwdaQrM

Chapter 2

State-of-the-art

Contents

This thesis focuses on the dexterous manipulation of deformable objects. In this chapter, we first review some basic knowledge related to this topic. It starts with an overview of the principles of visual servoing in case of robot interactions with rigid scenes in Section 2.1. Then, continuum mechanics is introduced in Section 2.2, as some of the methods thereafter mentioned in this manuscript rely on it. The remainder of the chapter presents a literature review of methods related to the dexterous manipulation of deformable objects. First, deformation tracking methods are presented in Section 2.3 because they are a key component when it comes to perform deformation servoing. Deformation servoing methods are presented in Section 2.4. Parameter estimation methods are presented in Section 2.5. Finally, the work accomplished during this PhD is positioned with regard to the existing literature in Section 2.6.

2.1 General overview of visual servoing

During this thesis, we proposed a method that relies on some principles of visual servoing to control the deformation of non-rigid objects using visual feedback. Thus, this first section gives a reminder on the basics in visual servoing, also known as visual servo control. Then, to illustrate some use-cases of visual servoing, some camera positioning tasks and robot interactions with rigid scenes are presented.

2.1.1 Basics of visual servoing

The following part is based on [CH06], which presents the basics of visual servoing. It focuses on the eye-in-hand case, i.e on the case where the camera is mounted on a robot manipulator or a mobile robot. Moreover, the camera is controlled in velocity. The aim of vision-based control is to minimize the error $\mathbf{e}(t)$ which is defined as follows:

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{im}(t), \mathbf{a}) - \mathbf{s}^* \tag{2.1}$$

where $\mathbf{s} \in \mathbb{R}^n$ represents the vector of n visual features, $\mathbf{s}^*$ represents the desired values of these features, $\mathbf{im}$ is the set of image measurements, such as points coordinates, and $\mathbf{a}$ is a set of parameters representing a potential additional knowledge about the system, such as the camera intrinsic parameters.

In the case of a motionless target, $\mathbf{s}^*$ is constant and the changes in $\mathbf{s}$ depend only on the camera motion. The matrix that relates the variation of the visual feature vector $\mathbf{s}$ to the

camera velocity $\mathbf{v}_c = (\boldsymbol{v}_c,\ \boldsymbol{\omega}_c) \in \mathbb{R}^6$ is called the feature Jacobian matrix[1] $\mathbf{J_s}$ and is defined as follows:

$$\dot{\mathbf{s}} = \mathbf{J_s}\mathbf{v}_c \tag{2.2}$$

Usually, the error is chosen to have an exponential decoupled decrease, i.e $\dot{\mathbf{e}} = -\lambda\mathbf{e}$. This choice leads to design the velocity controller $\mathbf{v^{ctrl}}$ as follows:

$$\mathbf{v^{ctrl}} = -\lambda\mathbf{J_s^+}\mathbf{e}, \text{ with } \mathbf{v^{ctrl}} = (\boldsymbol{v_{ctrl}},\ \boldsymbol{\omega_{ctrl}}) \in \mathbb{R}^6 \tag{2.3}$$

where $\boldsymbol{v_{ctrl}}$ is the instantaneous linear velocity vector of the velocity control law, $\boldsymbol{\omega_{ctrl}}$ is the instantaneous angular velocity vector of the velocity control law and $\mathbf{J_s^+}$ is the Moore-Penrose pseudo-inverse of the feature Jacobian matrix $\mathbf{J_s}$ such as $\mathbf{J_s^+} = \left(\mathbf{J_s^T}\mathbf{J_s}\right)^{-1}\mathbf{J_s}$.

In this quick introduction to visual servoing, two schemes are presented, as they are the bases for more complex servoing methods:

- the **Image-Based Visual Servoing (IBVS)** scheme, where $\mathbf{s}$ consists of features that are defined directly in the image data (e.g. pixel coordinates, image moments, etc)
- the **Position-Based Visual Servoing (PBVS)** scheme, where $\mathbf{s}$ consists of a set of 3D parameters estimated from image measurements (e.g. pose of an object, etc)

Image-Based Visual Servoing principles

In this section, the analytic expression of the feature Jacobian matrix $\mathbf{J_s}$ in the case of eye-in-hand manipulations is derived for a basic visual feature: the point.

Be ${}^c\mathbf{x_p} = (X, Y, Z) \in \mathbb{R}^3$ a 3D point whose coordinates are expressed in the camera Cartesian frame and $\mathbf{p} = (x, y) \in \mathbb{R}^2$ an image point that corresponds to its projection on the image plane of the camera, with x and y being respectively the horizontal and vertical image coordinates expressed in meters. The camera intrinsic parameters permit to relate ${}^c\mathbf{x_p}$ to its projected image point $\mathbf{p}$ thanks to the following set of equations:

$$\begin{cases} x &= X/Z = \frac{u - c_u}{f\alpha} \\ y &= Y/Z = \frac{v - c_v}{f} \end{cases} \tag{2.4}$$

where $(c_u,\ c_v,\ f,\ \alpha)$ correspond to the camera intrinsic parameters, with c_u and c_v the pixel coordinates of the principal point, f the focal length and α the ratio of pixel dimensions, and

1. The feature Jacobian matrix is also known as interaction matrix, denoted $\mathbf{L}_s$, or as image Jacobian matrix.

u and v are the horizontal and vertical pixel coordinates respectively.

A common choice in **IBVS** to construct the visual feature vector $\mathbf{s}$ is to use a set of image points $\{\mathbf{p}_1 \dots \mathbf{p}_\eta\}$, where η corresponds to the number of points used. When using a single image point as visual feature, the visual feature vector can be expressed as $\mathbf{s} = \mathbf{p} = (x, y) \in \mathbb{R}^2$. In this case, the image measurements $\mathbf{im}$ are the pixel coordinates of the image point $\mathbf{p}$, i.e. $\mathbf{im} = (u, v)$, and the additional knowledge vector is the camera intrinsic parameters, i.e. $\mathbf{a} = (c_u, c_v, f, \alpha)$.

The analytic expression of the feature Jacobian matrix $\mathbf{J_p}$ related to a single point $\mathbf{p}$ is well-known[2] and can be expressed as follows:

$$\mathbf{J_p} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \tag{2.5}$$

As one can notice, the feature Jacobian matrix $\mathbf{J_p}$ depends on the depth Z of the point relative to the camera Cartesian frame. Consequently, any control scheme that uses this type of visual feature must estimate the value of Z. Thus, it is not possible to use directly the feature Jacobian matrix $\mathbf{J_p}$ presented in Equation (2.5) to compute the velocity control law presented in Equation (2.3). Instead, an approximation of the feature Jacobian matrix $\hat{\mathbf{J}}_\mathbf{p}$ must be used, turning Equation (2.3) into:

$$\mathbf{v^{ctrl}} = -\lambda \hat{\mathbf{J}}_\mathbf{p}^{+} \mathbf{e}, \text{ with } \mathbf{v^{ctrl}} = (\boldsymbol{v}_{ctrl}, \boldsymbol{\omega}_{ctrl}) \tag{2.6}$$

where $\hat{\mathbf{J}}_\mathbf{p}^{+}$ is the Moore-Penrose pseudo-inverse of the approximation of the feature Jacobian matrix. Section 2.1.2 presents some methods to online estimate the feature Jacobian matrix.

To control the 6 **Degrees Of Freedom (DOFs)** of the camera, at least 3 image points are needed i.e. $\mathbf{s} = (\mathbf{p}_1^\top \ \mathbf{p}_2^\top \ \mathbf{p}_3^\top)^\top$. In this case, the feature Jacobian matrix $\mathbf{J_s}$ becomes:

$$\mathbf{J_s} = \begin{bmatrix} \mathbf{J}_{\mathbf{p}_1} \\ \mathbf{J}_{\mathbf{p}_2} \\ \mathbf{J}_{\mathbf{p}_3} \end{bmatrix} \tag{2.7}$$

However, when using only 3 image points, two problems occur: some configurations lead the feature Jacobian matrix to be singular [MR93] and four distinct indistinguishable camera poses for which the error vector is null exist [FB81]. Consequently, in practice we usually use more than three image points.

2. See [CH06] for instance for the full demonstration.

Over the years, complex **IBVS** methods have been proposed for robot interactions with rigid scenes, such as switching schemes [GH03], moving target tracking [PKK93] or hybrid visual servoing [CM00].

Position-Based Visual Servoing principles

In **Position-Based Visual Servoing** (**PBVS**), the visual feature vector $\mathbf{s}$ is defined using the pose of the camera with respect to a reference Cartesian frame. The computation of the camera pose requires additional knowledge about the system, such as the camera intrinsic and extrinsic parameters and the 3D model of the object with regard to which we want to position the camera. This knowledge is represented by the vector $\mathbf{a}$ in the expression of the error shown in Equation (2.1).

There exist several ways for defining a **PBVS** control scheme. Two options are presented here, to demonstrate that the motion resulting from the visual servoing differs depending on the choices about the problem formulation. First, let us introduce the notation that is used:

- $\mathcal{F}_c$, $\mathcal{F}_c^*$ and $\mathcal{F}_0$: current camera Cartesian frame, desired camera Cartesian frame and Cartesian frame attached to the object.
- ${}^c\boldsymbol{t}_0$ and ${}^{c^*}\boldsymbol{t}_0$: coordinates of the origin of the object frame $\mathcal{F}_0$ expressed relatively to the current camera Cartesian frame $\mathcal{F}_c$ and to the desired camera Cartesian frame $\mathcal{F}_c^*$ respectively.
- $\mathbf{R} = {}^c\mathbf{R}_{c^*}$: rotation matrix that gives the orientation of the desired camera frame relatively to the current camera Cartesian frame.
- $\mathbf{s}$ =($\mathbf{t}$, $\theta\mathbf{u}$): visual feature vector where $\mathbf{t}$ is a translation vector (defined below) and $\theta\mathbf{u}$ gives the angle/axis parameterization for the rotation.
- $\mathbf{J}_{\theta\mathbf{u}} = \mathbf{I_3} - \frac{\theta}{2}[\mathbf{u}]_\times + (1 - \frac{sinc\theta}{sinc^2(\frac{\theta}{2})})[\mathbf{u}]_\times^2$: feature Jacobian matrix related to the angle/axis parameterization for the rotation, with $\mathbf{I_3}$ the 3x3 identity matrix and $[\mathbf{u}]_\times$ the skew symmetric matrix associated to the vector $\mathbf{u}$.

Two possible options for a **PBVS** control scheme are the following:

1. $\mathbf{t}$ represents the translation vector of the origin of the object Cartesian frame $\mathcal{F}_0$ relatively to the current camera Cartesian frame $\mathcal{F}_c$ i.e $\mathbf{s} = ({}^c\boldsymbol{t}_0,\, \theta\mathbf{u})$.

   With $\mathbf{s}^* = ({}^{c^*}\boldsymbol{t}_0, 0)$, $\mathbf{e} = ({}^c\boldsymbol{t}_0 - {}^{c^*}\boldsymbol{t}_0,\, \theta\mathbf{u})$ and $\mathbf{J}_e = \begin{bmatrix} -\mathbf{I_3} & [{}^c\boldsymbol{t}_0]_\times \\ \mathbf{0} & \mathbf{J}_{\theta\mathbf{u}} \end{bmatrix}$,

the velocity control law is $\begin{cases} \boldsymbol{v}_c = -\lambda[({}^{c^*}\boldsymbol{t}_0 - {}^{c}\boldsymbol{t}_0) + [{}^{c}\boldsymbol{t}_0]_\times \theta\mathbf{u}] \\ \boldsymbol{\omega}_c = -\lambda\theta\mathbf{u} \end{cases}$

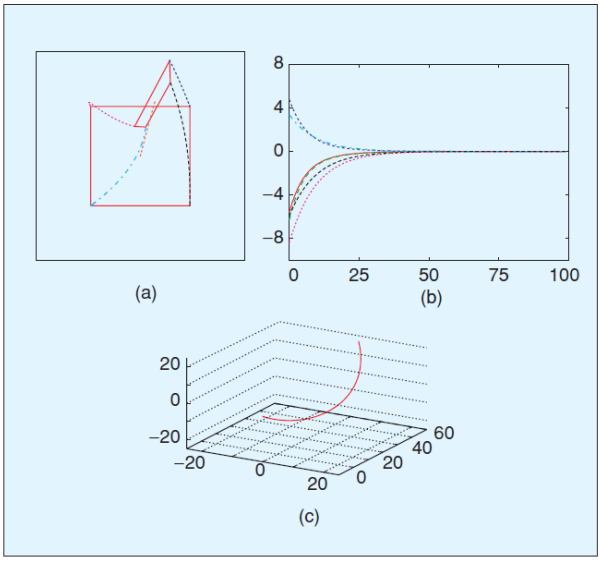
These choices result in a **PBVS** control scheme whose trajectory of the origin of the object frame follows a pure straight line in the image and whose camera 3D trajectory does not follow a straight line. For an illustration, see Figure 2.1a.

2. $\mathbf{t}$ represents the translation vector of the current camera Cartesian frame $\mathcal{F}_c$ relative to the desired camera Cartesian frame $\mathcal{F}_c^*$ i.e $\mathbf{s} = ({}^{c^*}\boldsymbol{t}_c, \theta\mathbf{u})$.

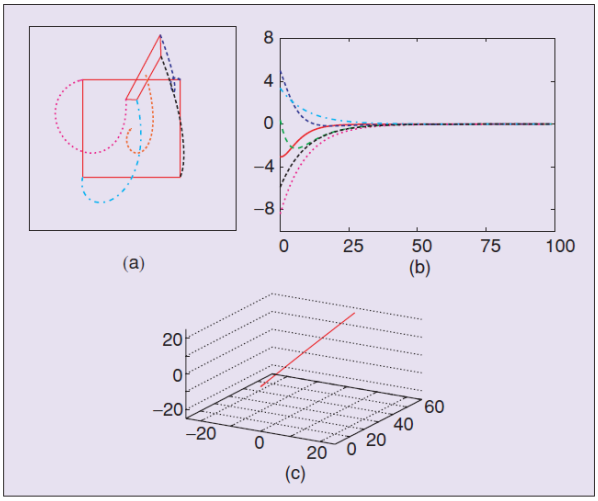
   In this case, $\mathbf{s}^* = \mathbf{0}$, $\mathbf{e} = \mathbf{s}$ and $\mathbf{J}_e = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\theta\mathbf{u}} \end{bmatrix}$,

   and the velocity control law is $\begin{cases} \boldsymbol{v}_c = -\lambda\mathbf{R}^{T}{}^{c^*}\boldsymbol{t}_c \\ \boldsymbol{\omega}_c = -\lambda\theta\mathbf{u} \end{cases}$

   These choices result in a **PBVS** control scheme whose trajectory of the origin of the object frame is not a straight line in the image and whose camera 3D trajectory is a straight line in the Cartesian space. For an illustration, see Figure 2.1b.



(a) With $\mathbf{s} = ({}^{c}\boldsymbol{t}_0, \theta\boldsymbol{u})$



(b) With $\mathbf{s} = ({}^{c^*}\boldsymbol{t}_c, \theta\boldsymbol{u})$

Figure 2.1: System behavior using four points as visual features and two different parameterizations. In each case, (a) = image points trajectories, (b) = $\mathbf{v}^{\mathbf{ctrl}}$ components (cm/s and dg/s) at each iteration of the control scheme and (c) = 3D-trajectory of the camera optical center expressed in the desired frame coordinates system. (from [CH06])

2.1.2 Numerical estimation of the feature Jacobian matrix: principles and examples

In visual servoing, the feature Jacobian matrix $\mathbf{J_s}$ is needed in order to compute the velocity control law that permits to reduce the user-defined error defined in Equation (2.1). The feature Jacobian matrix can either be analytically derived or numerically estimated. If the relation between the visual features and the **DOFs** of the system that are controlled is known beforehand, it is possible to analytically derive the feature Jacobian matrix $\mathbf{J_s}$. However, it is most of the time difficult to derive an analytic expression of the feature Jacobian matrix in **IBVS** as changes in the image are related to the depth point coordinate Z, as shown in Equation (2.5), which is usually unknown beforehand. Several methods have been proposed in order to numerically estimate online the feature Jacobian matrix. These methods are called **Uncalibrated Image-Based Visual Servoing (UIBVS)**.

Numerical estimation of the feature Jacobian matrix using the Broyden algorithm

The Broyden algorithm permits to iteratively update online the Jacobian matrix [Bro00]. Let $\mathbf{q}$ be the joint vector that gathers all the **DOFs** of the system that are controlled and $\mathbf{J_s} = \frac{\partial \mathbf{s}}{\partial \mathbf{q}}$ be the feature Jacobian matrix used to compute the velocity control law in order to reduce the user-defined error defined in Equation (2.1). The Broyden algorithm permits to iteratively update the feature Jacobian matrix as follows:

$$\hat{\mathbf{J}}_\mathbf{s}(k) = \hat{\mathbf{J}}_\mathbf{s}(k-1) + \varepsilon \frac{\Delta\mathbf{s}(k) - \hat{\mathbf{J}}_\mathbf{s}(k-1)\Delta\mathbf{q}(k)}{\Delta\mathbf{q}^T(k)\Delta\mathbf{q}(k)} \Delta\mathbf{q}^T(k) \tag{2.8}$$

where $0 < \varepsilon \leq 1 \in \mathbb{R}$ is a user-defined responsiveness gain, k is the current discrete time while $k-1$ is the previous discrete time, $\Delta\mathbf{s}(k) = \mathbf{s}(k) - \mathbf{s}(k-1)$ and $\Delta\mathbf{q}(k) = \mathbf{q}(k) - \mathbf{q}(k-1)$.

The Broyden algorithm has been used for instance in [SJ05]. In this paper, the authors propose an **UIBVS** method to achieve a primary task and a secondary task simultaneously. The primary task is a positioning task while the secondary task may be obstacle avoidance or singularity avoidance[3]. This method can only be employed if the number of **DOFs** of the primary task is greater than the number of visual features, because the secondary task will be accomplished using the **DOFs** not used for the primary task achievement to avoid that one task interferes with another. The authors explain how to set the weighted coefficient which

3. Singularity avoidance means that the system will avoid regions where lacking information to update the Jacobian matrix.

fixes the influence of the secondary task and how to formulate the gradient of the secondary task in order to achieve singularity avoidance and obstacle avoidance. The evaluation of their method consisted in controlling a 7-**DOF** robot in the context of singularity avoidance and the context of obstacle avoidance. The robot succeeded to reach its desired position in both cases.

Numerical estimation of the feature Jacobian using Least-Squares Minimization

Another method to numerically estimate online the feature Jacobian matrix is to use Least-Squares Minimization. Least-Squares Minimization permits to determine the feature Jacobian matrix that minimizes a user-defined error function relating this Jacobian matrix to available measurements [DL99].

Least-Squares Minimization has been used for instance in [SFJ10] to control a robot using an **UIBVS** scheme. The robustness of the feature Jacobian matrix with regard to outliers in the data used in the Least-Squares method is ensured thanks to the use of M-estimation [Hub81]. M-estimation is a statistical method that permits to reduce the sensitivity of the minimization of an objective function to outlier measurements.

First, a knowledge set $D = \{\mathbf{s}_p, \mathbf{q}_p\}_{p=1}^{P}$ of P pairs of visual features and joints configurations is built. Then, for a new pair $\{\mathbf{s}_c, \mathbf{q}_c\}$ an estimate of the feature Jacobian $\hat{\mathbf{J}}_\mathbf{s}$ is computed using the Iteratively Reweighted Least Squares[4] method to minimize the following cost-function:

$$\hat{\mathbf{J}}_\mathbf{s}(\mathbf{q}_c) := \underset{\hat{\mathbf{J}}_\mathbf{s}}{\operatorname{argmin}} \sum_{\mathbf{q}_k \in B_r(\mathbf{q}_c)} \rho\Big(\|(\mathbf{s}_c - \mathbf{s}_k) - \hat{\mathbf{J}}_\mathbf{s}(\mathbf{q}_c - \mathbf{q}_k)\| \Big) = \underset{\hat{\mathbf{J}}_\mathbf{s}}{\operatorname{argmin}} \sum_{\mathbf{q}_k \in B_r(\mathbf{q}_c)} \rho\Big(e\Big) \tag{2.9}$$

where $B_r(\mathbf{q}_c) := \{\mathbf{q}_p : \|\mathbf{q}_c - \mathbf{q}_p\| < r,\ p = 1, \ldots, P\}$ is an open ball of radius r centered around the joint configuration query $\mathbf{q}_c$ and ρ is the Geman-McClure M-estimator, which is defined as follows:

$$\rho(e, \sigma) := \frac{e^2}{e^2 + \sigma^2} \tag{2.10}$$

where $\sigma := 1.4826 \underset{i=1\ldots K}{median}\Big\{ |\|\mathbf{s}_c - \mathbf{s}_i\| - \underset{j=1\ldots K}{median}\{\|\mathbf{s}_c - \mathbf{s}_j\|\} \Big\}$ where the K pairs $\{\mathbf{s}_i, \mathbf{q}_i\}$ and $\{\mathbf{s}_j, \mathbf{q}_j\}$ are the K-nearest neighbors of the query $\{\mathbf{s}_c, \mathbf{q}_c\}$ in the joint space.

The memory is thereafter updated by registering the new pair $\{\mathbf{s}_c, \mathbf{q}_c\}$ in the knowledge set D.

The authors compared the Jacobian estimation error in the Frobenius-norm sense when

4. The Iteratively Reweighted Least Squares method is an implementation of robust M-estimation which can be used in nonlinear optimization problems.

using this method and when using the **Local Least-Squares (LLS)** proposed in [MSJ07], which is another **UIBVS** method. The effect of two types of outliers on these methods has been studied: the loss of visual features and the tracking inaccuracy. The proposed method outperformed the reference **Local Least-Squares (LLS)** method both in simulation and in practice when outliers occurred, while having a similar behavior when no outliers occurred.

2.1.3 Conclusion

This part briefly introduced the basics of visual servoing by presenting two visual servo control schemes, namely **Image-Based Visual Servoing** (**IBVS**) and **Position-Based Visual Servoing** (**PBVS**). Some methods based on these schemes were also presented. All of the aforementioned methods have been employed only for robot interactions with rigid scenes and positioning tasks. However, this thesis focuses on robot interactions with deformable objects. To this end, we use the core principles of adaptive visual servoing methods similar to the method proposed in [DL99]. The next part presents some principles of continuum mechanics, as they are a key component to the understanding of deformation servoing.

2.2 General overview of physics-based modeling of deformable objects

Continuum mechanics permit to describe the behavior of objects undergoing deformations due to external forces and constraints. This branch of mechanics notably describes how non-rigid objects deform when external loads are applied onto them. If a physics-based model of the object is known, a good prediction of the behavior of the object when subject to external loads can be obtained using continuum mechanics. Some deformation methods of the state-of-the-art being based on such models, a brief overview of the basics of continuum mechanics is presented in this section. Then, different ways of performing numerical simulations of deformable objects based on continuum mechanics are presented.

2.2.1 Continuum mechanics: modeling an object and its dynamics

Deformable objects can be described using continuum mechanics. Numerous books have been written on this topic, and the following paragraphs derive the formulae from [Thu97]. The *nature* of a deformable object hereafter means the way it behaves when it is subject to external

constraints: for instance a *linear elastic* object deforms linearly while a viscoelastic object resists partially to the deformation.

Let $\mathbf{x}_0(\mathbf{m}) \in \mathbb{R}^3$ be the rest position in the world Cartesian frame of a point $\mathbf{m} \in \mathbb{R}^3$ belonging to an object and that is expressed in the object Cartesian frame, and $\mathbf{x}(\mathbf{m})$ be the current position of this point. Let $\mathbf{u}(\mathbf{m}) = \mathbf{x}(\mathbf{m}) - \mathbf{x}_0(\mathbf{m})$ be the *displacement vector field*[5]. The displacement vector field $\mathbf{u}$, depicted in Figure 2.2, is important in continuum mechanics because it characterizes the overall deformation of the object. The relative elongation of the object is also known as *elastic strain* and denoted by $\overline{\boldsymbol{\epsilon}}$.

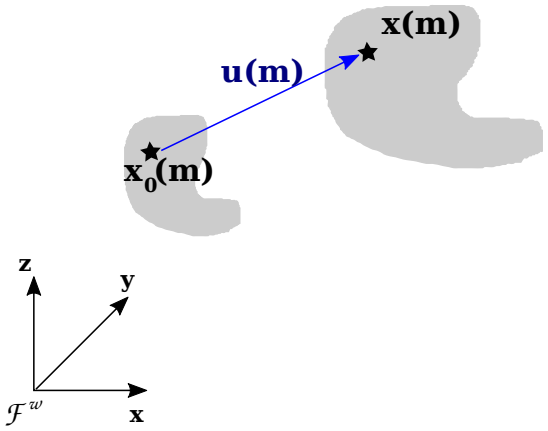


Figure 2.2: Illustration of the displacement vector field $\mathbf{u}$, where $\mathcal{F}^w$ corresponds to the world Cartesian frame.

Intuitively, it is possible to understand that a spatially constant displacement field does not induce strain. Consequently, the elastic strain $\overline{\boldsymbol{\epsilon}}$ is related to the spatial variations of the displacement vector field, i.e. to the gradient matrix of the displacement vector field $\bigtriangledown\mathbf{u} \in \mathbb{R}^{3,3}$. Two commonly used strain tensors are the Green's strain tensor $\overline{\boldsymbol{\epsilon}}_G$ and the Cauchy's strain tensor $\overline{\boldsymbol{\epsilon}}_C$ and are defined as follows:

$$\overline{\boldsymbol{\epsilon}}_G = \frac{1}{2}(\bigtriangledown\mathbf{u} + [\bigtriangledown\mathbf{u}]^T + [\bigtriangledown\mathbf{u}]^T \bigtriangledown \mathbf{u}),\ \overline{\boldsymbol{\epsilon}}_G \in \mathbb{R}^{3,3} \tag{2.11}$$

$$\overline{\boldsymbol{\epsilon}}_C = \frac{1}{2}(\bigtriangledown\mathbf{u} + [\bigtriangledown\mathbf{u}]^T),\ \overline{\boldsymbol{\epsilon}}_C \in \mathbb{R}^{3,3} \tag{2.12}$$

5. For the sake of brevity, hereafter the material coordinates $\mathbf{m}$ will be omitted.

The Cauchy's strain tensor is used to model linear elastic behaviors and is only valid for small deformations. However, it has the advantage of being fast to compute and is thus used for interactive time purpose. The Green's strain tensor permits to model nonlinear behaviors but is computationally expensive. Consequently, it is less commonly used in real-time implementations.

The *internal stress* $\overline{\boldsymbol{\sigma}}$, which is a symmetric tensor and expressed as a force per unit area, corresponds to the internal forces that occur in the object due to the strain it is subject to. Several constitutive laws exist to express the relation between the stress and the strain, depending on the nature of the object. One of them is the *Hooke's law*, which applies for linear elastic objects. A linear elastic object is an object for which the stress is linearly dependent of the strain. The hypothesis of linear elasticity holds only for small deformations[6]. In tensor notation, the *Hooke's law* can be written as:

$$\overline{\boldsymbol{\sigma}} = \overline{\mathbf{C}} \cdot \overline{\boldsymbol{\epsilon}} \tag{2.13}$$

where $\overline{\mathbf{C}}$ is the material tensor that is a rank four tensor which relates linearly the stress tensor coefficients to the strain tensor coefficients.

In the case of an isotropic material, $\overline{\mathbf{C}}$ depends only on the Young modulus E and Poisson's ratio ν and Equation (2.13) can be expressed as follows:

$$\overline{\boldsymbol{\epsilon}} = -\frac{\nu}{E} tr(\overline{\boldsymbol{\sigma}})\mathbf{I} + \frac{1+\nu}{E}\overline{\boldsymbol{\sigma}} \tag{2.14}$$

where $tr(\overline{\boldsymbol{\sigma}})$ represents the trace of the stress tensor $\overline{\boldsymbol{\sigma}}$ and $\mathbf{I}$ represents the identity matrix.

To understand what physically represents the Poisson's ratio, let us consider the deformable object shown in Figure 2.3, where l_1, l_2 and l_3 represent its lengths during resting state, h_1, h_2 and h_3 represent its lengths when the force $\mathbf{f}_1$ is applied onto its face $l_1 l_3$ of surface S_1. The *Poisson's ratio* ν links deformations of the sides l_2 and l_3 with the deformation of the side l_1 and is defined as follows:

$$\frac{h_2 - l_2}{l_2} = \frac{h_3 - l_3}{l_3} = \nu \frac{h_1 - l_1}{l_1} \tag{2.15}$$

Thus, the Poisson's ratio expresses the relation between the transverse strain and the axial strain exerted onto a deformable object. The Poisson's ratio represents the expansion or compression in the directions that are perpendicular to the forces directions.

6. In practice, it is often considered that a deformation is small if it is smaller or equal than 5% of the dimension of the object.

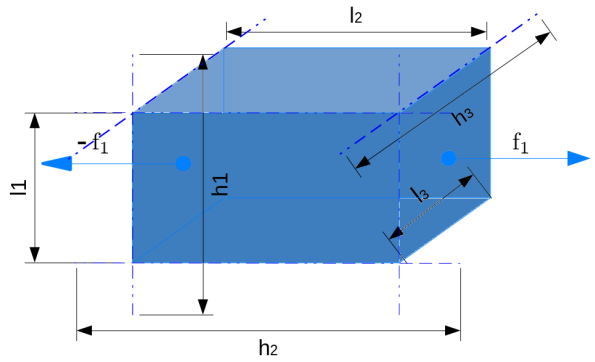


Figure 2.3: Deformable object submitted to a force $\mathbf{f}_1$ on its face $l_1 l_3$

The *Young modulus* E expresses the stiffness of the object subject to an uniaxial deformation and is defined as follows:

$$\frac{\mathbf{f}_i}{S_i} = E\Delta l_i, \text{ where } \Delta l_i = \frac{h_i - l_i}{l_i} \tag{2.16}$$

Figure 2.4 shows that a deformable object has an elastic behavior when the exerted force is low, i.e. it can retrieve its original form when the force stops. Then, it presents a plastic behavior when the force exceeds a threshold, i.e. the object will be irremediably deformed even when the force will have stopped. Finally, the breaking point indicates the amount of force that makes the object fracture.

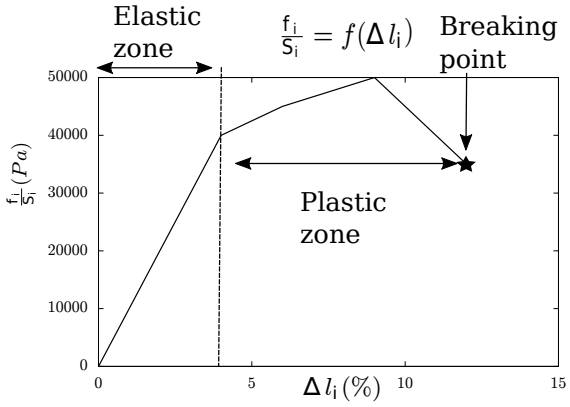


Figure 2.4: Relation of the ratio $\frac{\mathbf{f}_i}{S_i}$ with the transversal deformation Δl_i

The internal forces that a deformable object is subject to are directly related to the deformations by the physical properties of the object. The general equation of motion links the internal and external forces to which a deformable object is subject to with its displacement

vector field $\mathbf{u}$ in a dynamic case. The general equation of motion is defined by the second Newton's law and can be written as follows:

$$\mathbf{M}\ddot{\mathbf{u}} = f(\dot{\mathbf{u}}, \mathbf{u}) \tag{2.17}$$

where $\mathbf{M}$ is the mass matrix of the object, $\ddot{\mathbf{u}}$ is the second time derivative of the displacement vector field $\mathbf{u}$, $\dot{\mathbf{u}}$ is the first time derivative of $\mathbf{u}$ and $f(\dot{\mathbf{u}}, \mathbf{u})$ is a function that depends on the nature of the object and that can be represented by a set of **Partial Differential Equations (PDEs)**.

2.2.2 Finite Element Method: principles and numerical implementation

The **Finite Element Method (FEM)** [Nea+06] is a popular numerical method that permits to numerically solve continuum mechanics equations. The choice of the constitutive law used to describe the object defines how it deforms under applied forces. Depending on the constitutive law, different sets of physical properties (such as viscosity, elasticity ...) are used to describe the object. The **Finite Element Method (FEM)** transforms the **PDE** that results from the continuum mechanics equations into a set of algebraic equations that are solved numerically by discretizing the object in continuous connected volume elements. The discretized representation of the object is called a *mesh*. Different kind of elements can be used to model the object, such as tetrahedra or hexahedra.

Usually, **FEM** is used to determine the evolution over time of the rigid motion and the deformation of an object subject to known external forces. To do so, it is necessary to solve Equation (2.17) using a numerical integration scheme, such as the explicit, implicit and semi-implicit Euler integration schemes[7] presented for instance in [Nea+06].

Usually, the function f present in Equation (2.17) depends on the deformation the object is subject to. Let's take the particular case of an element e of a linear elastic object. At the static equilibrium, the deformation forces acting on the element vertices $\mathbf{f}_e$ are computed as follow:

1. The displacement field $\mathbf{u}_e$ permits to compute the elastic strain of the element $\overline{\boldsymbol{\epsilon}}_e$ using Equation (2.11) or Equation (2.12), which is then used to derive the stress field of the element $\overline{\boldsymbol{\sigma}}_e$ using Equation (2.13).

7. The explicit, implicit and semi-implicit Euler integration schemes are also called forward, backward and forward-backward Euler integration schemes respectively.

2. The deformation energy of the element E_e of volume V_e can be computed as follows:

$$E_e = \int_{V_e} \overline{\boldsymbol{\epsilon}}.\overline{\boldsymbol{\sigma}} dV \tag{2.18}$$

3. The deformation forces acting on the element vertices $\mathbf{f}_e$ are computed by derivating the deformation energy with regard to the nodes positions. Generally, this is a non-linear relation, but for a linear elastic object it can be linearized as follows:

$$\mathbf{f}_e = \mathbf{K}_e \mathbf{u}_e,\ \mathbf{f}_e \in \mathbb{R}^{3n_e} \tag{2.19}$$

where $\mathbf{K}_e \in \mathbb{R}^{3n_e,3n_e}$ is called the *element stiffness matrix* and n_e corresponds to the number of vertices of the element. The stiffness matrix depends on the displacement field u_e but it is omitted to ease the reading.

The deformation forces acting on the whole object are obtained by adding up the deformation forces acting on the vertices of each element e of the object mesh. This is done by assembling the *global stiffness matrix* of the whole body $\mathbf{K} \in \mathbb{R}^{3n,3n}$, with n the number of nodes that compose the mesh, as follows:

$$\mathbf{K} = \sum_e \mathbf{K}_e \tag{2.20}$$

where the element stiffness matrices are expanded to the size of $\mathbf{K}$ by filling in zeros at the positions corresponding to nodes not adjacent to the element.

Finally, in the static case the deformation forces acting on the whole object $\mathbf{f}$ are given by:

$$\mathbf{f} = \mathbf{K}\mathbf{u},\ \mathbf{f} \in \mathbb{R}^{3n} \tag{2.21}$$

This equation is also valid under the quasi-static assumption, i.e. when the object is deformed very slowly.

When the object is deformed in a dynamic situation, the linear algebraic equation of motion of a linear elastic object is used to derive the deformation forces, which are then given by:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}_{ext} \tag{2.22}$$

where $\mathbf{M} \in \mathbb{R}^{3n,3n}$ represents the mass matrix and $\mathbf{D} \in \mathbb{R}^{3n,3n}$ the damping matrix of the whole object, $\mathbf{u} \in \mathbb{R}^{3n}$ is the displacement field and $\mathbf{f}_{ext} \in \mathbb{R}^{3n}$ corresponds to the external forces

applied onto the object. It is important to notice that the motion corresponds to both the deformation of the surface and its motion at the same time.

In the case of a linear elastic object, the function f presents in Equation (2.17) is thus given by:

$$f\Big(\dot{\mathbf{u}}, \mathbf{u}\Big) = \mathbf{f}_{ext} - \mathbf{D}\dot{\mathbf{u}} - \mathbf{K}\mathbf{u} \tag{2.23}$$

Limitations of the linear FEM

The **FEM** generates a linear system of algebraic equations only if it is applied to a linear **PDE**. However, linearized elastic forces are valid only for small deformations because large rotational deformations induce highly inaccurate restoring forces. These inaccurate restoring forces generate a deformation of the mesh of the object, as shown in Figure 2.5.

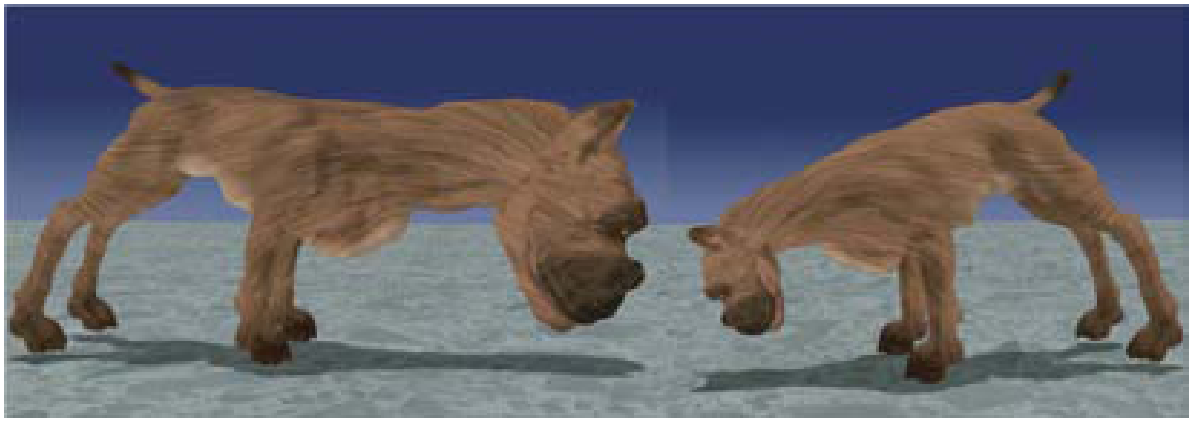


Figure 2.5: Illustration of the inaccurate restoring forces induced by large deformations in linear FEM. Left: Unrealistic deformed mesh due to large rotational deformations. Right: the original mesh , from [Nea+06]

Example of an alternative to the linear FEM: the corotational FEM

A popular method for avoiding inaccurate restoring forces due to large rotational deformations is called **corotational FEM**. It consists in extracting the rotational part of the deformation for each finite element, computing the forces with respect to the non-rotated frame and then putting back the object into the original rotated frame [MG04]. This solution is based on the linear continuum elasticity theory and uses **FEM** with linear displacement tetrahedra for solving the partial differential equations. The computation of the strain within each tetrahedron is done

using the Cauchy's linear strain tensor shown in Equation (2.12). Then, the stress is derived from the Hooke's law shown in Equation (2.13). In this case, $\mathbf{u} \in \mathbb{R}^{12}$ because a tetrahedron links four nodes with three **DOFs** so the material tensor depends only on the Young modulus and Poisson's ratio of the object if it is an isotropic material. In the proposed method, the equation of motion which governs the object displacement is the equation of motion assuming linearized elastic forces shown in Equation (2.22). A major advantage that motivated this choice is that the stiffness matrix $\mathbf{K}$ is constant and can thus be pre-computed once before the beginning of the simulation.

As said previously, the method proposed in [MG04] aims to solve the problem that linear elastic forces are not realistic and induce the unstability of the simulation when high rotational forces occur. To do so, they adapt the *Stiffness warping* method developed in [Mül+02]. Stiffness warping consists in computing the rotation of each vertex of the object with regard to a non-rotated frame. Then, it is possible to compute the linear elastic forces of each element into their non-rotated position. Finally, the object is turned back into its actual position. However, sometimes stiffness warping tends to induce ghost forces when the forces do not sum up to zero. The adapted method is done in several steps: (1) the rotation of each element with regard to a non-rotated frame is computed, ensuring that forces will sum up to zero; (2) knowing the rotation of each element, the whole object is turned into the non-rotated frame; (3) the linear elastic forces are computed for each element into the non-rotated frame, from the displacement between the original position in the non-rotated frame and the current position of the object in this frame; (4) the whole object is rotated back into its original position, and then the elastic forces of each element computed in the previous step are added up to know the global elastic force exerted onto the whole object.

The principle of this method is shown in Figure 2.6, where $\mathbf{x}$ represents the current position expressed in the rotated frame, $\mathbf{x}_0$ represents the original position of the object in the non-rotated frame and $\mathbf{R}_e^{-1}\mathbf{x}$ represents the current position of the object in the non-rotated frame. With this method, Equation (2.19) becomes:

$$\mathbf{f}_e = \mathbf{R}_e\mathbf{K}_e(\mathbf{R}_e^{-1}\mathbf{x} - \mathbf{x}_0),\ \mathbf{f}_e \in \mathbb{R}^{3n_e} \tag{2.24}$$

Finally, an implicit integration scheme, which is unconditionally stable, is used to transform the system shown in Equation (2.22) into a linear system.

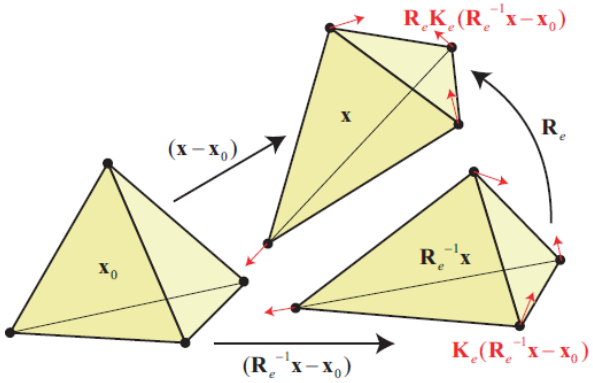


Figure 2.6: Illustration of the corotational **FEM** principle. $\mathbf{x}$ represents the current position expressed in the rotated frame, $\mathbf{x}_0$ represents the original position of the object in the non-rotated frame and $\mathbf{R}_e^{-1}\mathbf{x}$represents the current position of the object in the non-rotated frame, from [MG04].

2.2.3 Conclusion

This section reviewed the basics of continuum mechanics, as they are required to understand how a deformable object behaves when subject to external forces. The **FEM** has also been introduced as it is one of the most popular numerical implementations of the continuum mechanics laws. Next section introduces the notion of deformation tracking, which is required to permit robotic manipulation of deformable objects, and some related methods, some of them relying on the **FEM**.

2.3 Deformation tracking methods

Permitting robots to manipulate non-rigid objects requires to be able to track the evolution of the deformations they undergo. Different deformation tracking methods have been developed over the years. Each of them has its own advantages and drawbacks in terms of accuracy, computation time and computation payload. Additionally, the representation of the object might differ between the methods, which has an impact on the deformation features that can be used for robotic manipulations. Consequently, the choice of deformation tracking method, based on these criteria, is crucial when it comes to robotic manipulation of deformable objects. In this manuscript, we divide the deformation tracking methods into two categories: *model-free* methods do not rely on any physical model while *model-based* methods rely on physical models to describe either the object and/or its relation with its environment. In this section, some

model-free deformation tracking and deformable scene reconstruction methods are presented. Then, some model-based methods will be introduced. The advantages and drawbacks of each category are also discussed.

2.3.1 Model-free deformation tracking and deformable scene reconstruction methods

Model-free methods have the advantages of being easily transposed from one situation to another. However, due to the absence of models, it may be impossible to *track* the deformation but only to *reconstruct* it, i.e. to detect it without being able to give it a meaning. Some of the aforementioned methods are presented here.

Real-time progressive reconstruction of non-rigid scenes

[NFS15] presents a dense **Simultaneous Localisation And Mapping (SLAM)** system that permits to perform a progressive online reconstruction of a deformable scene using a **RGB-Depth (RGB-D)** sensor. The system requires neither any template nor prior scene model. First, the current frame is warped into a fixed, canonical frame. Once warped, the scene can be seen as rigid and standard **SLAM** updates can be used. Then, inverse warping transforms back the scene into the current frame. The method relies on the fact that surfaces move smoothly in space. Consequently, a sparse set of deformation nodes is used as bases to compute a dense volumetric warp function using interpolation. These deformation nodes are selected online by detecting surfaces where the nodes are further than a distance threshold of other deformation nodes. Combining the rigid transformations that transform the deformation nodes from the canonical frame into the current frame permits to handle surface deformation. A deformation-graph-based regularization term is added to increase the robustness of the method against noise and missing data due to occlusion.

The method was tested on live data. Experiments have shown that the method faces some difficulties in scenes that quickly move from closed to open topology or when large inter-frame motions occur.

Non-rigid 3D reconstruction without correspondences

[Sla+17] presents another real-time deformable scene reconstruction method using a single **RGB-D** stream. This method is geometry-driven but requires neither template nor shape priors. A volume of interest of the 3D space is discretized into voxels of chosen side length.

Then, a signed distance field assigns to each 3D point of the depth frame the signed distance to the center of the corresponding discretized voxel. The current frame is then warped into a canonical frame thanks to a minimization process that estimates the deformation field that will best align the signed distance field of the current frame with the one of the canonical frame. The cost-function that is minimized comprises: a term that ensures that neighbor points move in a similar manner while allowing the global motion not to be rigid; and a term that guarantees geometric correctness by forcing the conservation of the gradient magnitude of the signed distance field to unity. The method was validated on scenes where topology changes and large inter-frame motions occur.

Semi-automatic registration technique for Augmented Reality in surgical operations

In [Kim+12], the authors present a solution for marking and tracking a deformable surgical site using automatic detection of some image features in laparoscopic[8] surgery. This method does not rely on physics-based modeling but only on image processing. To do so, the surgeon needs to select a frame containing an organ of interest and to select the zone in which it lies. Then, the proposed system reconstructs its 3D surface on which the surgeon can indicate the surgical site location. Conformal mapping is used to project the 3D shape onto a plane preserving angles locally. Thus, neighboring points in the 3D shape also are neighboring points in the flattened plane, making possible to apply Pizarro and Bartoli's feature matching algorithm [PB12] to track the organ during the live streaming of the laparoscopic video. The algorithm permits to augment the laparoscopic video by adding some information (hidden structures of the organ, target highlighting...). Real experiments on an uterus surgical operation using laparoscopy were performed and the method presented good tracking performances.

Probabilistic-based registration method using Gaussian Mixture Model Expectation Minimization

[CB19] presents a probabilistic registration method based on Gaussian Mixture Model Expectation Minimization. To limit the impact of occlusion, visibility information is used not to consider occluded model points for the registration. In addition, the method uses **Coherent Point Drift (CPD)** regularization that permits to estimate a smooth spatial transformation of the model points, along with a Locally Linear Embedding regularization term that enforces topological consistency with regard to the original model. The method assumes that most

8. Laparoscopy is a diagnostic procedure for examining the organs inside the abdomen.

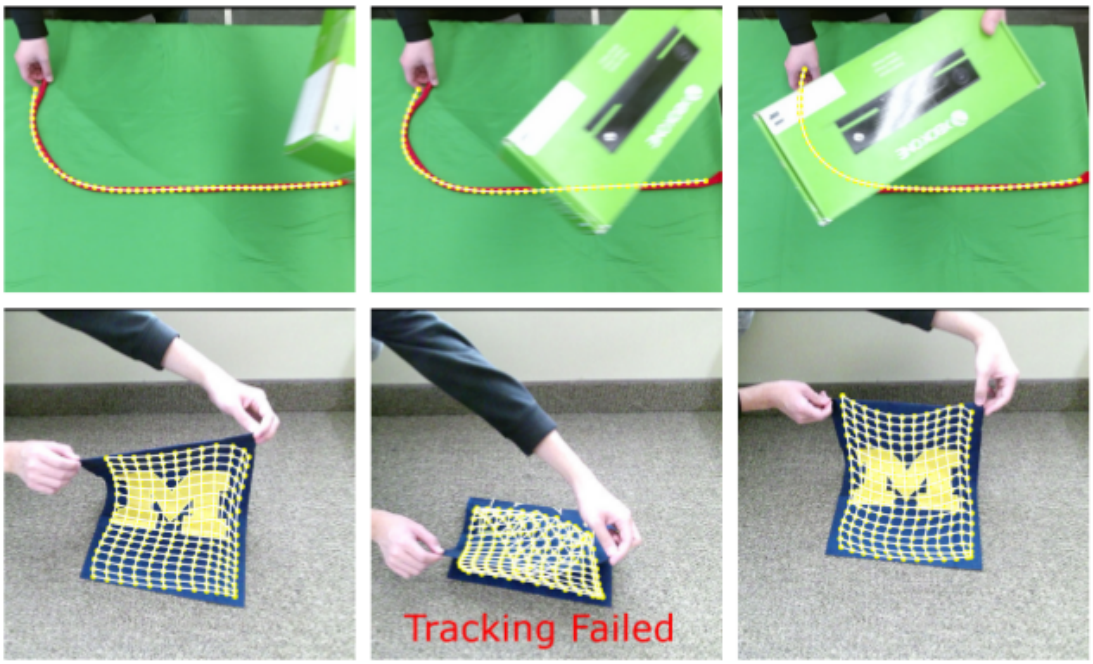


Figure 2.7: Some results of the method during different real data experiments. (Top) Evaluation of the method robustness against occlusion. (Bottom) Evaluation of the method robustness against tracking loss. Pictures from [CB19].

deformable objects are more resistant to stretching deformations. To enforce that the tracking respects this assumption, constrained optimization is used to allow bending and compression deformations while penalizing stretching deformations. When tracking failure is detected, a k-Nearest Neighbors template retry method is used to try to recover the tracking from previously faced states of the model. The similarity between the current point cloud and previously faced point clouds is computed using the 3D shape descriptor called **Viewpoint Feature Histogram (VFH)**. The recovery state that will be attributed to the current scene will be the one that corresponds to the recorded state having the closest **VFH** value. The method was evaluated against standard **CPD** and a combination of **CPD** and physics-based simulation on real and simulated data. Some results are illustrated in Figure 2.7. The method was more robust to occlusions but did not handle well delicate motions such as tying knots.

2.3.2 Model-based deformation tracking methods

Model-free deformation tracking methods are often more sensible to occlusions due to the difficulty of tracking recovery when lacking of additional knowledge. This section reviews some deformation tracking methods that use physics-based models to tackle this problem.

Deformation tracking in stereoscopic images using a mesh-free physics-based method

[Hao+14] presents a method for tracking the internal structures of the liver during robot-assisted procedures. This is a physics-based method not relying on a mesh of the organ of interest.

First, stereo-based tracking algorithm is used to determine the surface deformations. This algorithm is divided in three steps. The first step consists in reconstructing a sparse set of 3D feature points by triangulation from stereoscopic images. The second step is a **Moving Least-Squares (MLS)**-based surface reconstruction, which is a mesh-free, non-parametric method. The last step consist in Clustering-based filtering that creates a set of control points.

Once these steps are done, interpolation is used to precisely propagate surface motion to the internal structures, using a biomechanical model of the liver. A manual registration is done between the biomechanical model and the first pair of stereo images. Thereafter, a non-rigid registration is done by tracking the 3D control points and applying the external forces deduced from their motion onto the biomechanical model.

The authors tested their method by comparing its results with a Direct Tracking method proposed in [Hao+13] on in-vivo human liver surgery and ex-vivo porcine liver images. The results on the in-vivo experiment have shown that the proposed method decreases the projection error with respect to the Direct Tracking method. The results on the ex-vivo experiment have shown that the proposed method determines more precisely the position of manually added tumors in the liver than the Direct Tracking method.

Deformation tracking in 3D ultrasound sequences

[Roy+17] presents a method for real-time tracking of deformable structures in 3D ultrasound sequences. The target displacements are obtained by combining robust dense motion and mechanical model simulation. This approach has the advantage of providing correct motion estimation regarding different ultrasound shortcomings, such as speckle noise or large shadows.

First, the method consists in generating a 3D tetrahedral mesh model associated to the target. Then, the internal displacements are estimated thanks to a mass-spring-damper model. Elastic and damping coefficients are set on each spring. These coefficients and masses can vary locally, depending on soft-tissue homogeneities. Finally, the external displacements are estimated through an intensity-based minimization method. The cost function is a dissimilarity measure called Sum of Confident Conditional Variance. This dissimilarity measure is robust to global and local intensity changes caused by shadows presence or gain variation of the ultrasound imaging device. The method is illustrated in Figure 2.8.

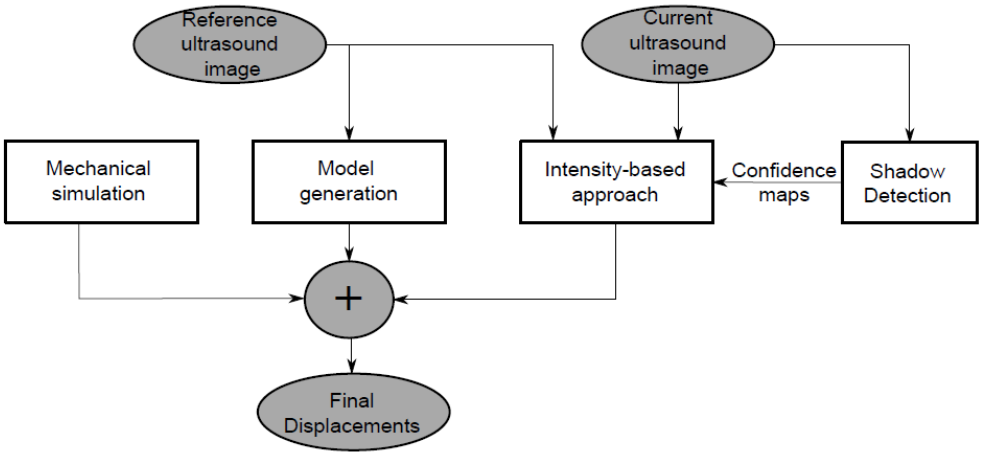


Figure 2.8: Workflow of the system, where the white squares represent the different steps of the method and the grey ellipses represent the input and output data, from [Roy+17]

The method was tested through different experiments. A first experiment has shown that the method is accurate for sequences including non-rigid motions and small confidence variations. A second experiment has shown that the method is robust during local and global intensity changes while other dissimilarity measures (such as Sum of Conditional Variance) are not. However, the proposed method can be inaccurate when the 3D mesh model goes out of the field of view or when the target undergoes high deformation (greater than 50%) regarding the provided elastic parameters.

Deformable object tracking using Corotational FEM

Some physics-based deformation tracking methods rely on a kind of models different from mass-spring models. For instance, the method proposed in [SKM19] uses Corotational **FEM** and a **RGB-D** sensor in order to track a deformable object. For each frame, a rigid tracking is first performed to determine the position of the object in the world Cartesian frame. Then, virtual forces are applied in a minimization scheme until the point-to-plane distance between the **RGB-D** sensor point cloud and the simulated object converges towards a minimum. This process is repeated each time a new frame is acquired. Thanks to this optimization process, this method requires neither the exact Young modulus and Poisson's ratio nor an exact mesh of the object to accurately track it. The method was compared to the method proposed in [Pet+18] using simulated data and showed a better independence with regard to the physical parameters used. The method was evaluated using real data with different objects, showing a mean point-to-plane error below the millimeter in each case. One of these experiments is

illustrated in Figure 2.9. During this experiment, a human operator deformed manually a pizza in front of the camera.

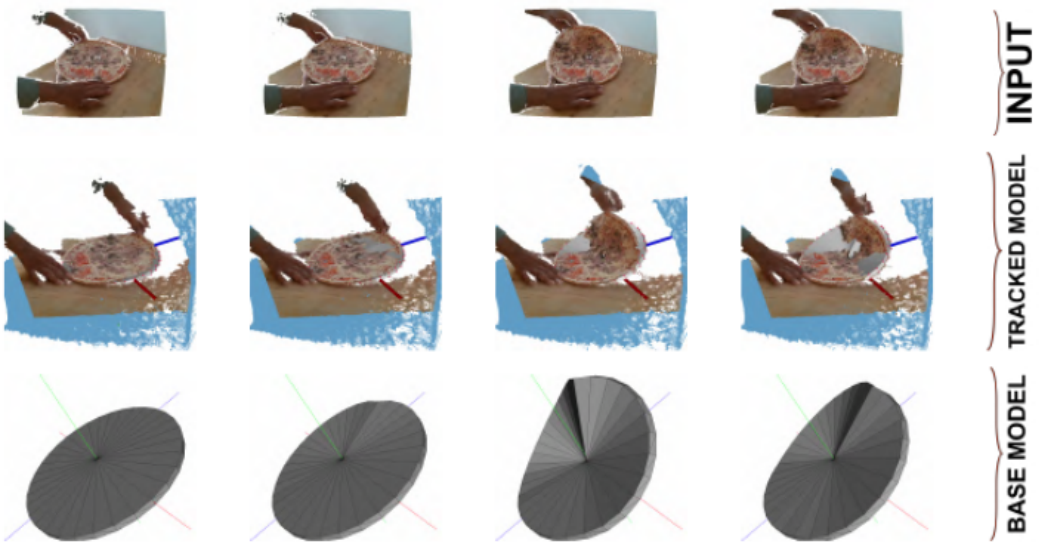


Figure 2.9: Some results of the method during a real data experiment consisting in tracking a deformed pizza, from [SKM19]. The input row designates the point cloud captured by the camera, the tracked model row presents the tracked model registered in the point cloud and the base model row presents the deformed model solely.

Heart tracking by registration between a FEM simulation and stereo endoscopic images

Another physics-based method relying on **FEM** simulation that can be used for image registration in the medical context of cardiac surgery is presented in [Pra+10].

A **FEM**-based biomechanical model of the heart is used. To ensure interactive rate simulation, the **FEM** is implemented using the **Total Lagrangian Explicit Dynamic** (**TLED**) algorithm presented in [Mil+07]. The mesh is generated from an initial scan phase followed by a manual segmentation. Finally, tetrahedralisation is performed on the surface mesh to obtain a volumetric mesh of the heart.

The system enables to perform spatiotemporal registration between the **FEM** mesh and the video sequence. To do so, the surface geometry of the heart is obtained using a stereo endoscope. Thereafter, the spatiotemporal registration is done through a least-squares distance optimization performed over one (or more) complete phase(s) of cardiac motion. The distance that is used is the distance between the projection of the 3D fiducial positions of the **FEM** mesh and the corresponding points recovered from the stereo video sequence. Finally, a forward **FEM**

simulation makes the mesh nodes follow the observed surface of the heart. Since the simulation time step is smaller than the video frame rate, linear interpolation is applied over the duration of the frame.

The authors tested their method by comparing its 2D overlay accuracy when the positional constraints are combined with the recovered force constraints with the results of a manual frame-by-frame video feature tracking. The results have shown that the combined constraints offer a good trade-off between the positional-only and force-only alternatives. Moreover, the accuracy of this method and its ability to work at interactive rate make it suitable for dynamic guidance during interventions.

Deformable tracking via registration between RGB-D images and a Finite Element model

Another example of deformation tracking algorithm used for robot servoing is presented in [Pet+17b]. A humanoid dual-hand robot manipulates a pizza dough. To fulfil this task, real-time tracking of the deformable dough is needed. The model-based deformation tracking method that is employed is inspired from the work presented in [PLS15].

The deformation tracking method begins with a segmentation step that permits to consider solely the points of the point cloud that belongs to the object. The segmentation is performed on the RGB frame and the corresponding points of the point cloud are retrieved using the known correspondence between the RGB frame and the point cloud. The segmentation method is based on graph cuts where the problem is formulated as an energy minimization problem. To ensure a temporal coherence of the segmentation between successive frames, the contour of the object is extracted from the previous segmented image. The energy minimization is performed only on the points that are close to the contour. It permits to limit the risk of having outliers inside or outside of the object. This segmentation step is illustrated in Figure 2.10.

The deformation tracking is done by registration of a corotational FEM model of the object with the segmented point cloud. The registration is performed in several steps. First, a rigid **Iterative Closest Point (ICP)** algorithm is used to register the visible vertices of the surface mesh extracted from the previous frame with the segmented point cloud. Then, external forces exerted by the point cloud on the mesh are derived using geometrical point-to-point distance between the pairs of visible surface mesh vertices and point cloud points. However, forces attracting the contours are weak due to the fact that only the points lying on the contours of the mesh are attracted to the extended area in the point cloud. To solve this problem, the authors propose to weight the external forces, giving greater weights to the points near the

(a) Original frame. (b) Segmentation. (c) Distance map. (d) Trimap.

Figure 2.10: Illustration of the segmentation step of the method, from [Pet+17b]. The original RGB frame (a) is first segmented (b). The distance map (c) permits to determine the points of the contour of the object that will be used during the energy minimization. These points are depicted in grey in (d)

contours of the object mesh. Finally, a conjugate gradient method is used to estimate the deformation of the object.

The authors tested their method on simulated data, real data and during a robotic manipulation experiment. The robotic manipulation consisted in controlling a humanoid dual-hand robot to manipulate a silicon pizza dough. An example of tracking results during such a task is presented in Figure 2.11. The robot had to follow a trajectory commonly performed by pizza chefs that makes the pizza dough change from one hand to the other. The deformation tracking process permitted to achieve this complex task. However, the process is put in standby when the pizza was flipped because the occlusions became too numerous. Meanwhile, the rigid **ICP** process was maintained.

Registration on point clouds using a probabilistic method

[TT18] presents a probabilistic framework called **Structure Preserved Registration (SPR)** to perform deformable wire tracking using Gaussian mixture model registration on point clouds. To limit noise sensitivity, occlusion due to obstacles is detected and the Gaussian centroids that are considered to be invisible are not updated. The local topology is captured and assumed to be almost constant over time, to ensure the integrity of the tracked wire. The **CPD** method is used to track the rigid displacements of the wire while allowing deformations to occur. The results of the **SPR** are thereafter used in physics-based simulations to ensure the correctness of the tracking according to physical constraints. Consequently, some knowledge of the physical properties of the object is required.

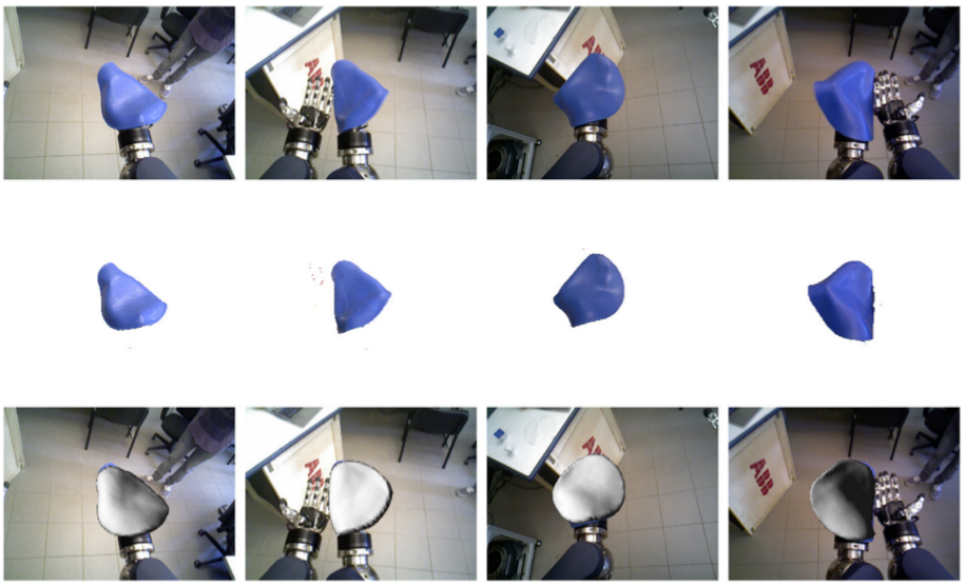


Figure 2.11: Illustration of some tracking results of the method during a robotic manipulation, from [Pet+17b]. The first row presents the RGB stream. The second row presents the segmentation results. The third row presents the results of the registration of the tracked model with the RGB frame.

Sparse Linear Elastic Shape-from-Template deformation tracking method

A novel method of Shape-from-Template tracking that handles deformations generated by sparse forces has been proposed in [MH17]. To do so, it relies on the Projective Elastic Space Property that encodes both the traditional reprojection constraint and the elastic model of the object. The reprojection constraint is a constraint that enforces the fact that the deformation field, computed using the physical model, applied on the rest model of the object and projected on the image plan matches the observations. The deformation tracking is performed thanks to a minimization scheme that enforces that the sparse deformation force vector applied on the object has the smallest possible number of components different from zero while respecting the reprojection constraint.

The method was tested on real and simulated data on object for which the thickness is negligible with regard to the maximum side length of a single triangular element. The method was compared with other Shape-from-Template methods from the state-of-the-art: a linear iterative method, a non-linear iterative method, a sequential method based on Kalman Filtering and an analytic method. For the simulated data, the magnitude and orientation of the sparse deformation forces are randomly generated. The reconstruction accuracy is measured in terms of the ratio of the number of force non-zero components correctly recovered to the total size

of the force vector and in terms of 3D residual error between the generated deformed 3D points and the reconstructed ones. The sparse force recovery is exact when the number of non-zero components does not exceed 45% of the total size of the force vector. However, this rate decreases if the noise on the observations increases. The method outperformed the state-of-the-art methods. During the real data experiments, a stereovision system was used to obtain both the observations and the ground truth. On four datasets, the proposed method outperformed the state-of-the-art methods, reaching a 1mm accuracy in terms of mean 3D residual error.

2.3.3 Conclusion

This section reviewed different methods of deformation tracking. These methods can be employed to control a robot such as it changes the shape of an object. Indeed, tracking the deformation permits to build an error function used for the control law. This topic is covered in the next section.

2.4 Deformation servoing methods

A survey presenting recent progress of deformable objects sensing and manipulation in domestic and industrial robotic applications can be found in [San+18]. A new categorization of deformable objects is proposed, which takes into account both their geometry and their physical properties. The authors classify the robotic applications according to the type of deformable objects that they permit to manipulate.

In this section, we chose a complementary approach for the categorization of the presented methods. Indeed, one of the objective of this thesis is to propose novel deformation servoing methods to address some of the challenges arising from robotic manipulations of non-rigid objects. The non-rigid objects can also be the end-effector itself, since soft robots become more affordable thanks to 3D printing for instance. We also want to evaluate the accuracy of *model-free* deformation servoing methods, that do not rely on any physical models, against *model-based* methods, that rely on physical models of either the object and/or its environment. Consequently, this section first presents model-based methods and finally focuses on model-free methods.

2.4.1 Model-based methods

Two-phase shape modeling of elasto-plastic objects

In [HYK11], the authors propose to use a four-element model to approximate the dynamic characteristics of object deformations. This model is decomposed into an elastic part and a plastic one. The strategy consists in two phases, illustrated in Figure 2.12.

The first one is the parameter estimation phase, during which the robot is controlled in position. The control input corresponds to the desired position to avoid over-deformation of the object. The contact force is measured while the deformation is kept constant for a certain duration. Then, the pressure is released. At the end of the releasing step, the object might still be deformed due to its plastic parameters. It makes possible to identify its viscoelastic and plastic parameters. Consequently, it is possible to identify its viscoelastic and plastic parameters.

The second one is the shaping phase, during which the robot is controlled in force. The estimated viscoelastic and plastic parameters permit to deduce the force required to obtain the desired shape of the object.

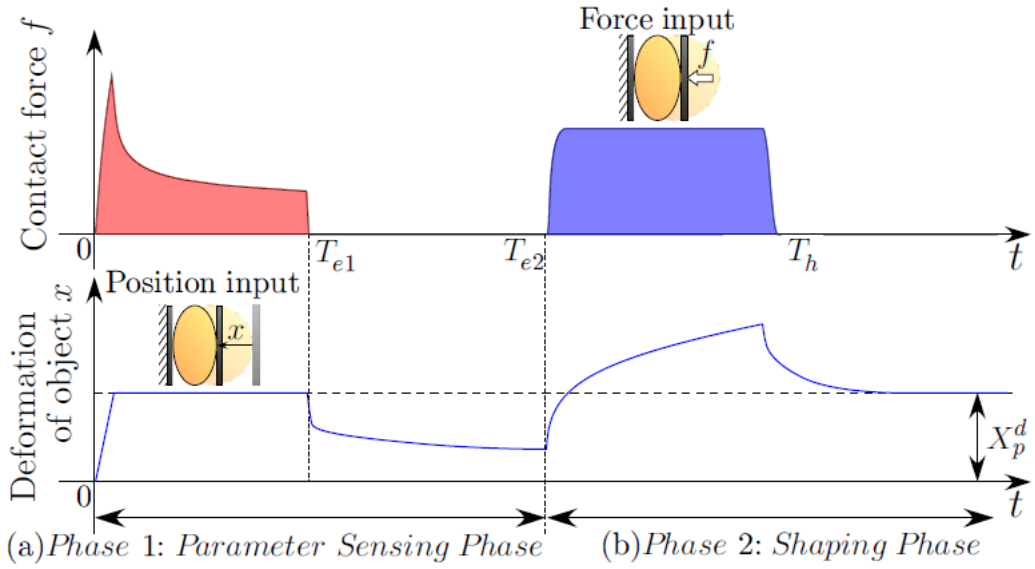


Figure 2.12: Two-phase strategy for generating the desired shape. In the first phase, the unknown viscoelastic parameters are estimated by applying a position control to the robot, the final shape being the input. In the second phase, the final shape is obtained by controlling the robot in force to manage the plastic behavior of the object. Figure from [HYK11]

The experimental system is composed of a linear encoder for controlling the robot in po-

sition, a load cell for controlling the robot in force and a marker applied on the object that can be detected by a vision sensor to measure the object deformation. The method was able to estimate the parameters of a parallelepipedic object made of wheat clay and to apply a definitive deformation of a few millimeters in 40s.

Two-dimensional shape modeling using several manipulators

[DS11] presents a solution using several manipulators to change the shape of a deformable object by a finite number of actuation points. The initial actuation points are supposed to be given to the system.

A mass-spring-damper model is used. The objects are considered to be both viscoelastic and plastic. They are represented by a Voigt model[9] for the viscoelastic properties in series with a damper for the plastic properties. The objects are supposed to be planar. Consequently, the shape error is two-dimensional. This shape error is a discrete representation of the difference between the desired positions of the set of actuation points and the actual ones. Consequently, a null error does not ensure that the whole surface of the object matches the desired shape.

The controller is designed by minimizing an energy-like criterion derived from the shape error between the desired and initial shapes. A robust controller is derived from this controller using robust control theory to overcome problems arising from parameter uncertainty. All the manipulators have their own controller without communication between them.

Two sets of simulations were performed to demonstrate the abilities of the system. The first set consisted in applying the shape controller to deform an initial circular shape into either an ellipse or a square with a different number of actuation points. The results have shown that the more actuation points there are the more the accuracy of the system improves. The second set of simulations consisted in handling a deformable object with a three-finger robotic hand without deforming it. The robust controller succeeded to accomplish this task.

Grasping of deformable objects using a multi-fingered robotic hand

A method that uses a non-linear mass-spring model to represent the object deformations due to the mechanical loads generated by the robotic hand is presented in [Zai+17]. The method handles tangential friction forces such as slipping and sticking effects. The non-linear mass-spring model permits to determine the required contact forces and deformations to grasp with a minimal squeezing the object. These simulated forces are thereafter used in a force control

9. A Voigt model is a two-element model made of a spring and a damper in parallel.

strategy to actually achieve the grasping with the real robot and object.

The non-linear mass-spring model is used to determine the initial grasp configuration using a geometrical criterion. Once this configuration is determined, a position-control scheme is used to move the fingers of the robotic hand towards their desired position, which ensures contact with the object while not deforming it. The determination of the contact forces that permit to grasp the object with a minimal squeezing is an iterative process. The first step consists in detecting the facets of the object mesh that are in contact with the fingers. A finger is in contact with a facet if the distance between the projection of its fingertip and the center of the hemispherical fingertip is below a threshold. Thereafter, the friction and normal contact forces are evaluated. The normal forces are proportional to the elasticity and damping parameters of the object, and to the penetration distance between the fingertip and the facet surfaces. The friction contact forces due to sliding are computed by evaluating the relative velocity between the fingers and the object. The sticking friction contact forces are computed using the elasticity and damping parameters of the object. A criterion comparing the sticking forces and the normal forces permits to determine if the contact is sticking or slipping, and to set the tangential friction contact forces accordingly. Finally, the grasping stability is evaluated by checking if a static equilibrium has been reached. A static equilibrium is reached if the normal and tangential friction forces balance the weight of the object. A principle schematic to illustrate the method is depicted in Figure 2.13.

Experiments were conducted on deformable objects having different shapes and made of different materials. The robotic hand is a Shadow hand installed on a Kuka LWR4+ robot. The hand is equipped with a BioTac tactile sensor that permits to determine normal forces. Compression tests were performed to evaluate the elasticity and damping properties of the objects. The method successfully managed to grasp all tested deformable objects.

FEM-based path planning for shape preservation

In [Fra+11], a **FEM** simulation is used for effector path planning while preserving the shape of a deformable object. The method uses probabilistic roadmaps with an A^* graph search technique for motion planning. During an offline step, a graph is generated. Its nodes are collision-free configurations and its edges correspond to the cost of moving from one node to another. The cost function takes into account the distance between robot configurations and the deformation cost estimated using a **FEM** simulation tool. During the online path planning, new deformation costs are inferred from learning ones using Gaussian Process regression.

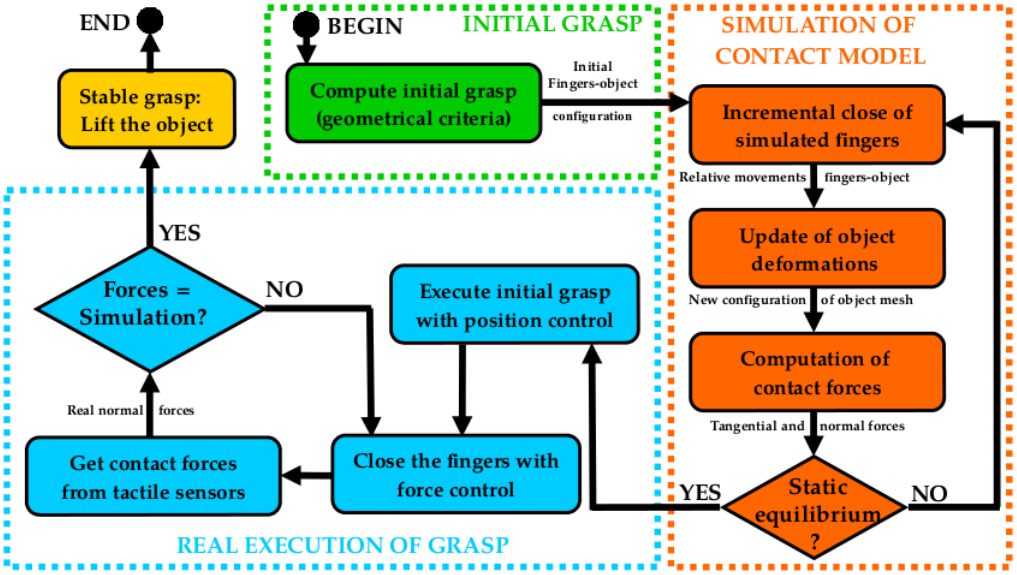


Figure 2.13: Principle schematic of the method, from [Zai+17].

Pose control of a deformable object using force sensor

A method to perform pose control of a deformable object held by a gripper, using only a force sensor and a physical model of the object, has been proposed in [San+20]. A Recurrent Neural Network observer is used to estimate the contact forces exerted by a gripper on the deformable object from the force-torque sensor. The estimated contact forces are used with the physical model of the object to estimate its deformation. Then, the pose of the object is obtained by computing the centroid of the object and estimating its position in Cartesian space along with its orientation. Finally, an error vector term is defined from the difference between the current and desired pose of the centroid, and used to compute the joint velocity command using the known robot Jacobian matrix. Both the deformation sensing and the deformation control were evaluated using the odometry of the robot to get the ground truth pose of the held object.

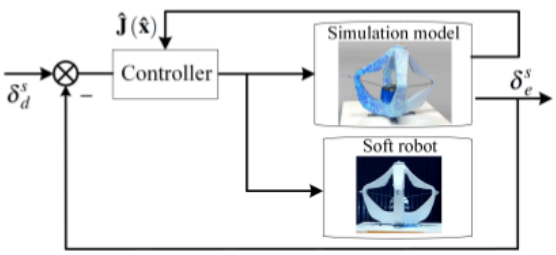
Deformation servoing can be used to control the deformations of non-rigid end-effector(s) instead of to controlling rigid end-effector(s) interacting with deformable objects.

FEM-based soft robot control

FEM has been used for instance in [Zha+17] for controlling soft robots. The control of soft robots is a particular case of control of deformation, the goal being to maintain a particular shape for the robot in order to achieve a task. The proposed method combines a **FEM** physics-based-simulation controller with a **PBVS** controller in a closed-loop control. This method

uses a switched controller consisting of: an open-loop controller using **FEM** simulation to compute the Jacobian matrix needed for the control design, a closed-loop controller that uses the simulation-based Jacobian estimation and visual servoing and switching conditions between the two loop controllers. The two controllers are depicted in Figure 2.14 The visual servoing ensures that the simulated robot and the actual one are in the same configuration. The closed-loop controller is employed when the image feature is available. Otherwise, the system switches and uses instead the open-loop controller. The method was validated using a soft robot driven by four cables.

(a) Open-loop controller.



(b) Closed-loop controller.

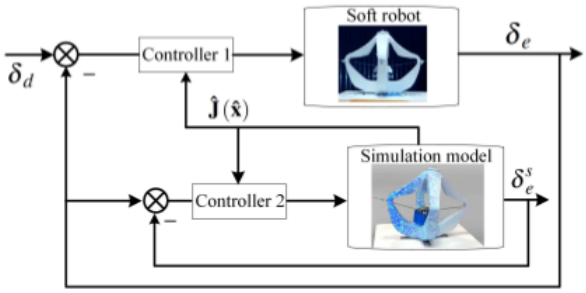


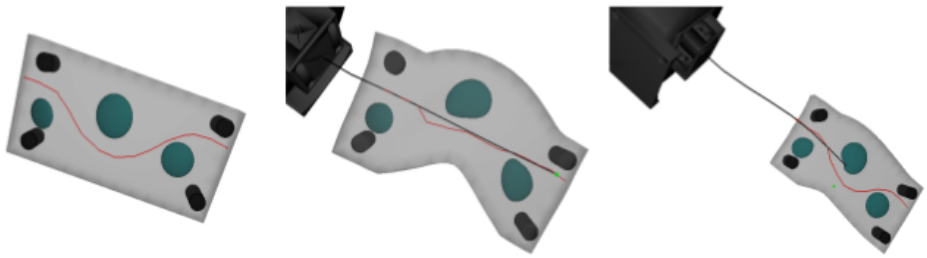
Figure 2.14: Principle schematic of the two controllers, from [Zha+17].

Deformation servoing can also be employed when both the end-effector(s) and the object that needs to be manipulated are deformable.

FEM-based automatic flexible needle insertion

FEM physics-based simulation has also been used for automatic flexible needle insertion in deformable tissue using a predefined path [Ada+16]. The method relies on the quasi-static assumption. Consequently, the deformation is performed very slowly by waiting that the system made of the object and the needle reaches an equilibrium before going further. The method is based on minimizing a cost function that penalizes the distance between the tip of the needle and the desired trajectory, the insertion angle and the angle between the tangent of the trajectory and the needle tip. The simulation is used for two purposes. The first purpose is to predict the needle tip position and the object configuration at the current time step knowing the current forces. The second purpose is the computation of the Jacobian matrix of the aforementioned cost function with regard to the robot joint configuration, the needle tip position and the object configuration. This Jacobian matrix permits to compute the change of robot joint configuration needed to reach the needle tip position that minimizes the trajectory error.

The method sensitivity, accuracy and computation were evaluated using simulation only, i.e. without implementation on an actual robot. The result of such an experiment is depicted in Figure 2.15. Using a naive approach that considers that both the gel and the needle are rigid, the automatic needle insertion failed while the proposed approach successfully fulfilled the task. The method can still work in real-time for a 1800-node simulation, with a mean error lower than 5mm in both a scenario with only one obstacle and a scenario with several obstacles.



(a) Initial Trajectory (b) Our Approach (c) Naive approach

Figure 2.15: Results of a simulated experiment of robotic needle insertion, from [Ada+16]. In green are depicted obstacles while red lines correspond to the desired trajectory. The naive approach considers that both the gel and the needle are rigid and failed to fulfill the task, as shown in (c).

2.4.2 Model-free methods

A disadvantage of model-based deformation servoing methods is that they require a lot of prior knowledge about the system, such as the object mass, geometry or physical parameters. Moreover, model-based methods are often expensive in terms of computation time. To avoid these drawbacks, one can decide to use a model-free method instead. In this section, we present model-free methods that rely only on visual information, since they are the closest to what we intend to propose.

Model-free deformation servoing based on visual information

[Nav+14] is a first example of model-free method to control a robot for deforming an object. In this article, the authors present two adaptive methods that require neither the prior identification of the object deformation model nor the camera intrinsic/extrinsic parameters. In both case, the deformable object is supposed to be quasi-static and purely elastic. A fixed camera provides the visual feedback to the controller in real-time.

The first solution they propose is a real-time estimation of the deformation Jacobian matrix. This solution is done in two steps. The first step consists in designing a desired motion command in deformation coordinates. To do so, the system estimates the deformation Jacobian matrix through the minimization of the deformation error. The second step consists in mapping the control actions to the end-effector velocities using the estimated Jacobian matrix.

The other solution is a combination of a minimum of offline information with online adaptation of variable parameters. This solution does not require numerical differentiation, but requires to acquire some test points before the control experiments. During the collection of test points, the end-effector displacement vectors and the visual feedback vectors must not be colinear. These vectors permit to construct a regression matrix before the control experiments. This regression matrix remains constant during the online deformation test and is used to build an estimated projection model (see Figure 2.16). Finally, the parameter adaption rule is driven by a combination of two error terms. The first term is the error between the test points current position and their desired one. The second term is the error of the estimated projection model.

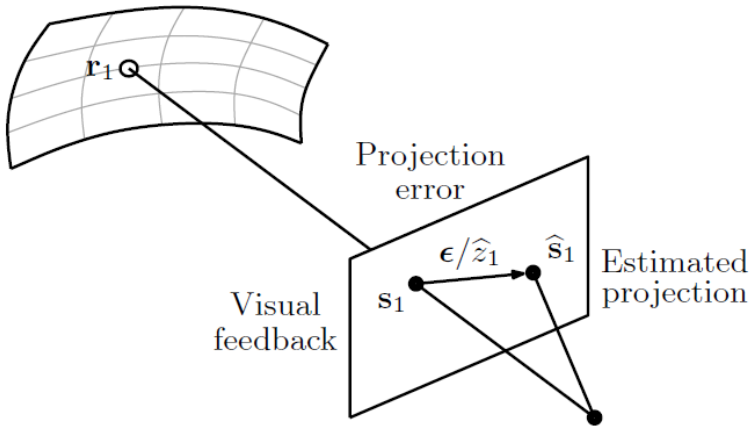
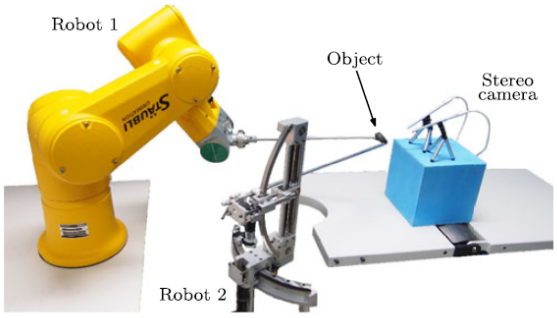


Figure 2.16: Illustration of the projection error, from [Nav+14].

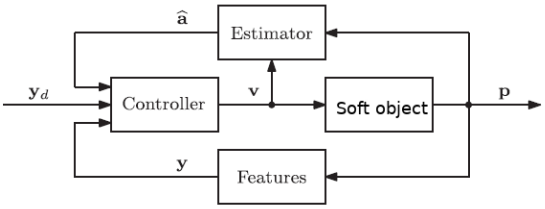
The method was tested using a robot manipulator controlled by a joint velocity law. Some markers were placed on the deformable object for the visual feedback. The deformation features were two-dimensional, such as the curvature of a beam-like foam object or the desired position of a point of interest. Both controllers succeeded to perform the desired tasks. However, the results have shown that the controllers suffer some limitations, such as the need of filtering the optical flow for the first method due to noise and the fact that the second controller assuming linear deformations reduces its optimal application to small deformations.

Simultaneous positioning and shaping using image-based visual servoing

In [Nav+16], the authors present an adaptive method to automatically change the shape of an unknown elastic object without needing its deformation model. To do so, the method uses a velocity controller and an algorithm that estimates in real-time a deformation feature vector dividing the servoing task into a positioning task and a shaping task. A stereo camera observes some deformable feature points whose 3D-locations are used to construct the deformation feature vector. An approximation of the deformation model is iteratively computed by asymptotically minimizing the model matching error between the deformation model and the velocity of the feature points. The hardware setup and the work-flow graph are presented in Figure 2.17.



(a) Hardware setup



(b) Work flow graph

Figure 2.17: Overview of the system presented in [Nav+16].

The method was tested using two manipulators. The error used to compute the control law is the difference between the target shapes and the actual ones. Figure 2.18 shows the results with different types of targets. This method presents better results than a fixed model but suffers some limitations. For example, it is not suitable for mono-camera systems, it requires a good choice of the feature points and it badly handles small curvature deformations.

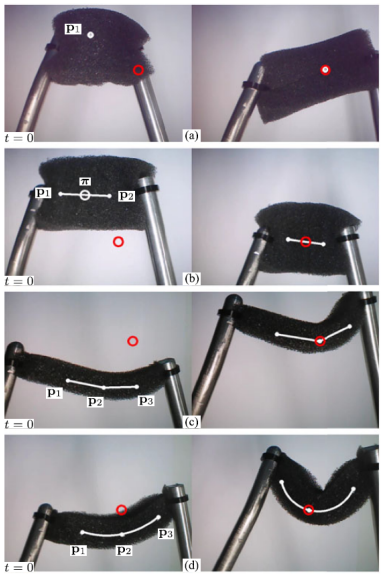


Figure 2.18: Results for different shape control terms: (a) explicit feature point position control term, (b) compression distance with centroid, (c) folding angle with feature point and (d) normalized curvature with feature point , from [Nav+16].

Fourier-based model-free 2D deformation control

[NL18] presents a method to tackle overall shape deformation problem. Contrarily to the previous methods, this one only control the 2D shape of the object. To do so, a static camera observes the object while it is manipulated by the robot. The contour of the object is extracted and expressed as a parametric curve with regard to some defined origin. The parametric expression of the 2D contour of the object is transformed into a compact vector of Fourier coefficients. The desired and current parametric curves are estimated in local regions by taking a few sampling points. The effect of a change in the robot motion on the current parametric curve is observed to estimate a Jacobian-like matrix. This matrix can thereafter be used in a Jacobian-based control law to locally minimize the shape error. This shape error is defined as the difference between the current parametric curve Fourier coefficients and the desired parametric curve ones. Experiments on an actual robot to deform a real object have shown that a simple binary image is sufficient to compute the contour local parametric curves and that the method is able to work in real-time. However, it has also shown that the object needs to have a good contrast with the background. Moreover, the system falls in a local minimum configuration if the desired contour is not reachable or if the number of Fourier coefficients is not sufficient to correctly describe the shape of the object.

A Robust Data-Driven Approach for Online Learning and Manipulation of Unmodeled 3D Heterogeneous Compliant Objects

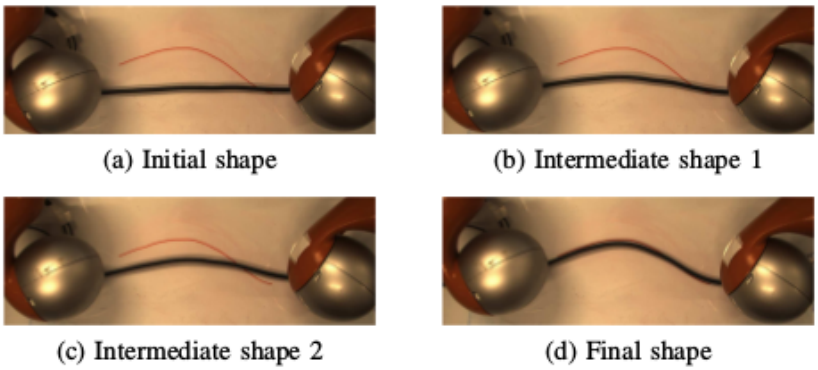
A method that permits to perform indirect positioning tasks of a compliant object has been proposed by Alambeigi *et al.* in [Ala+18]. The object is grasped by one or several manipulator(s). The positioning of the object is performed by making match the image projection of some of its surface points, designated as feature points, to desired image points locations. The method does not require any prior knowledge on the object and is able to handle changes of mass during the manipulation task. The Jacobian matrix that relates the grasping points motions to the feature points motions is estimated online using the Broyden update rule. Using the updated deformation Jacobian matrix, a constrained optimization problem is solved to determine the robot control law permitting to deform the object according to the desired position of the feature points. The experimental evaluation was performed on a deformable object filled with several pieces. The method was tested in different conditions: with one manipulator using one feature point for the positioning, the same experimental condition with a change of mass occurring due to cutting the surface of the object, with two manipulators and two feature points and finally the same than the previous experiment with change of mass occurring.

Linear deformable objects are widely spread in the industry. For instance, we can quote wires that are numerous in electronic devices. Consequently, proposing methods designed to tackle the particular problem of handling such objects is of importance.

Wire shape control using a dual-arm robot

A method to control the 2D shape of a wire using two effectors has been proposed by Zhu *et al.* in [Zhu+18]. Each effector has 3 **DOFs**: two for translations and one for rotation. The wire shape is approximated by a Fourier series. The geometric deformation model is adapted online, assuming that a change in the Fourier coefficients depends linearly on a change in robot position. Sets of changes in Fourier coefficients and sets of changes in robot position are continuously acquired. These sets are used in a least-squares minimization scheme to compute the Jacobian matrix that relates the robot motions to the changes of shape. However, if the robot did not move in all the affordable directions, the Jacobian matrix estimation update is delayed to avoid any singularity. The Jacobian matrix is finally used to compute the control law. The method was tested using a wire held by two grippers. One of these experiments is depicted in Figure 2.19. The goal was to reach a desired 2D shape painted in red on a white background. One can notice that the method successfully managed to converge towards the

desired 2D shape.



(a) Initial shape (b) Intermediate shape 1

(c) Intermediate shape 2 (d) Final shape

Figure 2.19: Results of a 2D wire shape control experiment, from [Zhu+18]. The red line depicts the desired 2D shape to reach. (a) and (d) show respectively the initial and final shapes. (b) and (c) show some intermediate shapes that are reached during the manipulation.

2.4.3 Conclusion

Table 2.1 presents a summary of the deformation servoing methods presented in this section. This table contains the data they require, the input parameters, the hypotheses, the need of an offline step and the deformation abilities. We can notice that most of the methods that permit a global control of deformation of the object rely on physical models or only control the 2D shape of the object. Most of the methods of the literature require an offline step to be performed.

Table 2.1: Summary of the deformation servoing methods

Method	**Measurements**	**Required parameters**	**Hypotheses**	**Offline step**	**Type of deformation**
HYK11	Robot force and position, 3D position of the object	4 scalars	Quasi-static assumption	YES	Local
DS11	Contact points position	Mass, 3 spring constants, 2 damping coefficients	None	YES	Global
Zai+17	Robot force and position	Non-linear mass spring model	None	YES	Global
Zha+17	Robot configuration using computer vision	3D model, Young modulus, Poisson's ratio, mass, switching parameter	Quasi-static assumption	NO	Global
Fra+11	Robot trajectory, 3D position of object feature points	Deformation weighting cost, 3D model, elastic properties	Straight-line motions	YES	Global
Ada+16	Effector basis position, penetration angle, 3D positions of sampling points	3D model, Young modulus, Poisson's ratio, mass, friction coefficient, switching distance threshold	Quasi-static assumption	NO	Global

Continued on next page

Table 2.1 – continued from previous page

Method	Measurements	Required parameters	Hypotheses	Offline step	Type of deformation
San+20	Robot force	3D model, Young modulus, Poisson's ratio, mass	Linear elasticity	YES	Local
Nav+14	Robot velocity, 2D visual features	n projection parameters, n feedback scalars, 2 coefficients	Small deformation, quasi-static assumption	YES	Local
Nav+16	3D positions of feature points	Positive saturation scalars, first order filter gain	Quasi-static assumption	YES	Local
NL18	Robot configuration, feature points position	Number of Fourier coefficients, saturation threshold, size of the region	Quasi-static assumption	YES	Global
Zhu+18	Robot velocity, segmented RGB frame	Number of Fourier coefficients, saturation threshold, size of the region	Quasi-static assumption	YES	Global
Ala+18	Robot configuration, feature points position	Broyden's algorithm parameter, error threshold	None	NO	Local

2.5 Parameter estimation methods

Knowing the physical parameters of the objects permits to predict their deformation behavior. Knowing the physical parameters of the objects might come handful to manipulate deformable objects, and is required when one wants to use a model-based deformation servoing method. Different ways of identifying physical parameters have been developed. Deformable objects are most of the time heterogeneous. Indeed, they deform differently depending on the direction they are stimulated. However, capturing all the deformation parameters of a heterogeneous material can be complicated due to the high number of samples needed and the size of the resulting set of parameters. This section first presents some parameter estimation methods assuming a homogeneous behavior. Finally, some methods handling heterogeneity are presented.

2.5.1 Homogeneous objects

Elasticity parameter estimation of homogeneous isotropic objects

[Fra+10] presents a method to determine the elasticity parameters of a homogeneous isotropic object using a **RGB-D** camera and a robot manipulator equipped with a force-torque sensor.

First, the algorithm creates a volumetric tetrahedral mesh of the object of interest. To do so, several point clouds are captured from different points of view and turned into a triangular surface mesh. Then, a registration technique based on the **ICP** algorithm permits to reconstruct the volumetric mesh. Figure 2.20 illustrates the geometry acquisition process. If only a rough estimation of the elasticity parameters is sought, it is possible to take a unique observation of the undeformed object and to assume that the hidden faces are planar.

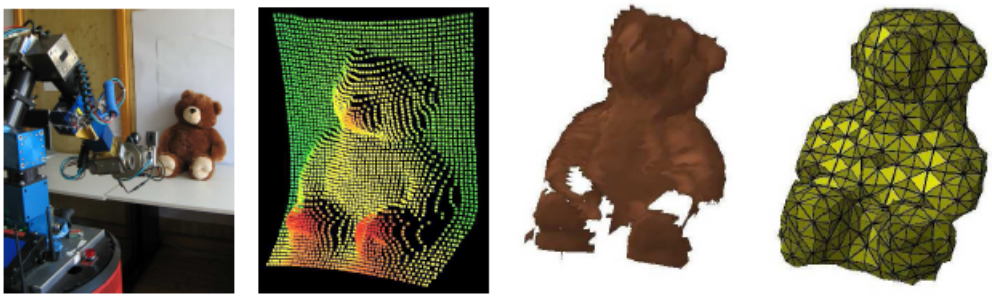


Figure 2.20: Object reconstruction: (a) the robot observes the object, (b) a point cloud is obtained from the range-camera, (c) a surface-mesh is reconstructed from different point clouds, (d) a tetrahedral mesh is computed from the surface mesh, from [Fra+10].

Thereafter, the identification of the elasticity parameters E and ν is performed by recording different applied forces and the resulting displacements. To limit occlusion, the robot manipulator is equipped with a thin stick to probe the object. A **FEM**-based simulation of the object is run to reproduce the observed deformation. The simulated object is subject to the same forces than the actual object and variations on its elasticity parameters are applied by a gradient-descent optimization process. This process aims at minimizing an error function to estimate the elasticity parameters. Let M be the undeformed object model, $\mathbf{z}^f$ be the set of force measurements, $\mathbf{z}^s$ be the corresponding surface mesh and $\mathbf{p}$ be the vector of contact points. The error that is minimized is the Euclidean distance between the object surface when deformed by the robot manipulator and the simulated deformed surface:

$$Err(E,\nu) = dist(simulate(E, \nu, M\ , \mathbf{z}^f, \mathbf{p}), \mathbf{z}^s)$$

Figure 2.21 shows the results of an experiment conducted on an inflatable balloon whose tetrahedral mesh was made of 815 elements. The average time for computing its optimized elasticity parameters was of 192.5 seconds and different experiments conducted to similar Young modulus values, indicating a homogeneous deformation behavior of the object.

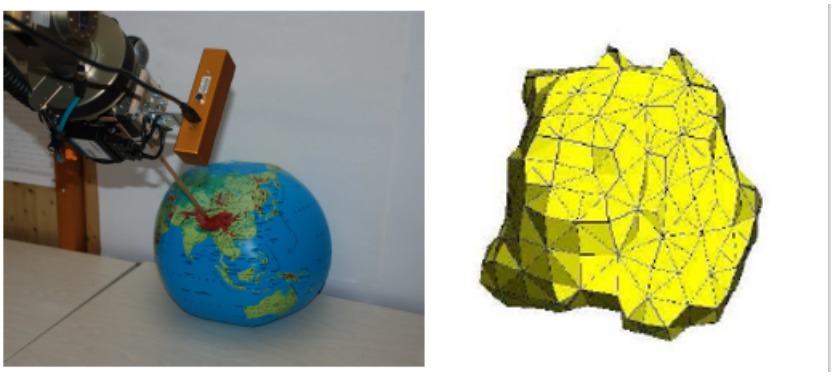


Figure 2.21: Estimation of the parameters of an inflatable balloon: (left) real experiment, (right) tetrahedral mesh obtained after elasticity parameter estimation, adapted from [Fra+10]

Determination of the viscoelasticity parameters of soft tissues

Elastic properties are not the only ones that can be estimated. In [SSB06], the authors propose a method using a haptic device and a force sensor to estimate the viscoelasticity parameters of a soft tissue. This soft tissue is used in a real-time **FEM** simulation with haptic feedback.

A viscoelastic material has an instantaneous elastic response when subject to a force, which can be modeled by the Hooke's law. This elastic response is followed by a slow and continuous change in response at a decreasing rate. Consequently, the simulation of a viscoelastic material requires considering the loading history of each element for the displacement computation at each time step. The method estimates the system steady-state elastic response, the two first elastic coefficients of the instantaneous elastic response and the two first relaxation times related to damping. To do so, it performs curve fitting on the experimental force relaxation data.

From the experimental force relaxation function, it is possible to determine the Poisson's ratio of the object and the normalized value of elastic moduli, which is the ratio between the average of the two first elastic coefficients of the instantaneous elastic response and the system steady-state elastic response.

Precomputed data are used to generate correct approximations of force and displacement responses of the object between two time steps of the simulation to feed the haptic thread.

The method was tested by performing the same compression tests on a 3D cube with the proposed method and with the ANSYS[10] finite-element package. The viscoelastic model estimated by the method matched the one of ANSYS. Moreover, the precomputation approach accuracy was compared to the direct solution and resulted in an error smaller than 0.01 millimeter for nodal displacements of a few millimeters. However, the method consumes a huge memory storage space because of the need of recording two datasets for each node of the model.

Estimation of the rheological properties of uniform and layered food products

Another example of method estimating not only the elasticity parameters of an object is presented in [WH11]. Indeed, the authors propose a method for estimating the **rheological**[11] properties of uniform or layered food products. It is done using a dynamic Finite Element model based on a five-element physics-based model. These five elements are two Kelvin-Voigt elements, which are a spring in parallel with a viscous element, and one viscous element connected in series. During the experiments, the authors used a 2D model with triangular elements. However, it can be extended to a 3D model by replacing the triangles by tetrahedra and adding Z-axis components to all the vectors and matrices used in the method.

The estimation of the rheological properties is done by performing an inverse Finite Element optimization. First, the object is compressed and the force responses are recorded using a tactile tensor. Then, Finite Element simulations are performed, tuning the rheological parameters to

10. ANSYS is an engineering simulation software. Interested reader may find more information in [Flu09].
11. A rheological object has both elasticity parameters and plastic properties.

minimize the difference between the experimental data and the simulated data. However, two sets of parameters are needed: one for describing the force responses (called force parameter set) and one for describing the deformation (called shape parameter set).

To validate their method, the authors used two simulations, one with the force parameter set and one with the shape parameter set. For the simulation using the force (respectively the shape) parameter set, only the force responses (respectively deformations) were kept. Finally, the results of the two simulations are "merged" to obtain an accurate simulation of the reality. Simulated data of the compression of several Japanese sweets showed a good similarity with data obtained from real experiments.

Rheological parameters estimation through indentation tests and inverse analyses on FEM

Forging has been interested in estimating the parameters of metal pieces for a long time to ensure good quality. For metal structures, parameter estimation can be made by performing test simulations and inverse analyses on **FEM**. In [BM11], **FEM** is first used for simulating the behavior of an isotropic rheological object with different sets of parameters containing the Young modulus and the yield stress. It permits to generate a curve with the corresponding measurable quantities. Then, **Proper Orthogonal Decomposition (POD)** and **Radial Basis Functions (RBF)** are used for: reducing the number of quantities to observe for identifying unknown parameters of the actual object; and interpolating parameters that were not tested from measured quantities. In this paper, only "measured" quantities of a simulated indentation test are used for testing the method, but actual measured quantities could be used. The measured quantities, real or simulated, are used in an optimization **Trust Region Algorithm (TRA)** algorithm to find the interpolated set of parameters which is the closest from the observed one.

Application of the elasticity parameter estimation to remotely estimate contact forces

The estimation of the elasticity parameters of a deformable object can be used to permit remote estimation of contact forces as proposed in [Pet+17a]. The overall method relies on a corotational **FEM** model of the deformable object. The deformation tracking method proposed in [PLS15] is employed to track the deformation the object undergoes. The deformation results from the fact that the object is probed by an end-effector equipped with a force-torque sensor. When a steady-state of deformation is reached, the elasticity parameters of the object are

estimated thanks to the gradient-free Nelder-Mead optimization method. The contact forces measured by the force-torque sensor, whose location in the object frame is supposed to be known, are applied on the vertex that corresponds to the location of the probing in a corotational **FEM** simulation. Changing the elasticity parameters of the simulation results in a different final steady-state. The error function that is minimized with regard to the elasticity parameters in the optimization process is a linear function of the sum of two point-to-point Euclidean-distance-based error terms. The combination of these two error terms permits to limit the impact of wrong correspondences due for instance to occlusion. For each new evaluation of the error function, the simulation is reset in its resting state.

Once the estimation of the elasticity parameters of the object converges, the estimated values can be used to perform remote contact force estimation. A gradient-free Nelder-Mead optimization method is used to estimate the contact forces required to maintain the object model in the same deformed configuration than the one that is observed.

The elasticity parameter estimation and the remote force estimation were evaluated on real data. The experimental setup consisted in a stuffed toy compressed by a robotic manipulator equipped with a force-torque sensor while a **RGB-D** sensor observed the scene. For the elasticity parameter estimation, no ground truth was available but the different experiments converged to the same value. For the evaluation of the remote force estimation, the ground truth was available thanks to the force-torque sensor. The method was able to estimate contact forces relatively close to the ground truth measurements.

2.5.2 Heterogeneous objects

The previous part focused on methods based on the homogeneity assumption. On the contrary, the following methods take into account the heterogeneity of the object.

Ultrasound image-based visual servoing applied to deformation tracking

The authors of [PK17] present a robot-assisted system using an ultrasound probe to obtain the elastic information of a moving tissue. This system permits to move the ultrasound robot such as its motion cancels the unwanted one of the observed tissue.

First, the method uses a dense non-rigid motion estimation in 2D to perform motion tracking on the moving soft tissue. The goal is to estimate the deformation parameters of an initial **Region Of Interest (ROI)**. This is done by minimizing a dissimilarity function, which depends on the selection of the warping function that warps the image points using the deformation

parameters. It is important to maintain the position of the **ROI** stable even when the tissue is moving to ensure that the solution is robust to non-rigid motion induced by the tissue deformation. To do so, an ultrasound image-based visual servoing is used to synchronize the probe motion with the tissue motion.

Then, a robotic arm equipped with a 6-axes force-torque sensor permits to obtain the pre- and post-compression states of the tissue by applying a varying force along the axial direction of the ultrasound probe using a force control scheme. Finally, the elastogram[12], which gives the strain estimation, is the result of the convolution of a Least-SQuared (LSQ) strain filter with the axial displacement map.

The method was tested by manually applying motions to an ultrasound phantom and observing if a hard tissue target corresponding to an artificial cyst remained visible in the center of the elastogram. This experiment has shown the robustness of the system to perturbation motions.

Elasticity parameter estimation of heterogeneous object using probing

The elasticity of an object can be expressed using another parameterization than the Young modulus and Poisson's ratio. The method presented in [Bic+09] uses a probe equipped with a force sensor to estimate the Lamé's coefficients of heterogeneous deformable objects, such as human face tissue. The data acquisition uses a marker-based stereovision system with three camera sensors. These sensors are placed in a triangular manner to minimize occlusion due to probing.

The parameter estimation is done through an elastostatic **FEM**-based simulation able to perform non-linear material modeling. The non-linearity is possible by locally characterizing the stress-strain function with a linear corotational **FEM** at sampling positions. The estimation of the parameter vector at each sampling position is done by incremental loading of known forces while measuring the resulting displacements, followed by a minimization process. Then, at an arbitrary point, surrounding sample parameter vectors are used to non-linearly interpolate the parameter vector thanks to **Radial Basis Functions (RBFs)**.

Figure 2.22 shows the results of the system on a heterogeneous pillow whose mesh is made of 1,691 tetrahedron. The deformations can be displayed at interactive frame rates.

The authors highlight several limitations of their method. For instance, viscosity and plasticity are not taken into account. Neither is the frictional behavior of the object, as only forces

12. An elastogram is the strain map of a tissue.

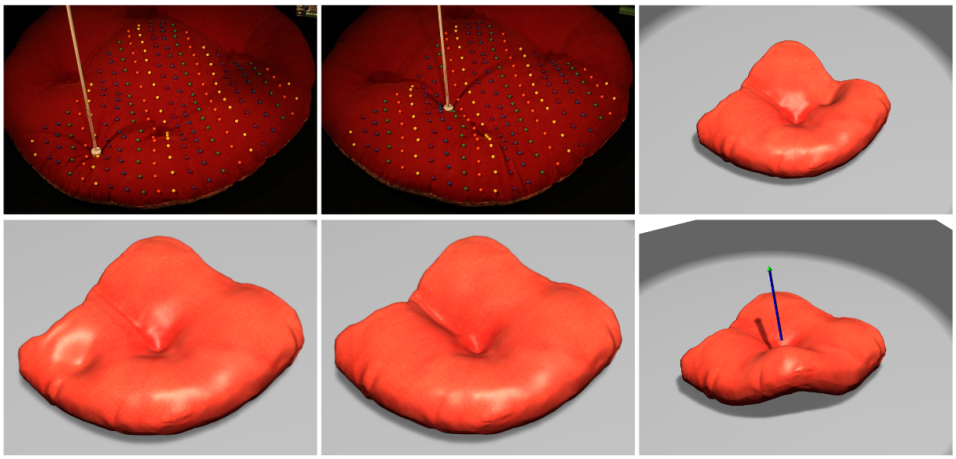


Figure 2.22: (Left and middle) Comparisons of captured and synthesized deformations for a heterogeneous non-linear pillow. (Right) Interactive deformations of the model produced by pushing (top) and pulling (bottom), from [Bic+09].

in the direction of the probe are measured. Finally, the minimization process can fall in a local minimum, leading to wrong deformation synthesis.

Non-destructive methods for elasticity and plasticity parameter estimation in forging context

Some parameter estimation methods handling heterogeneous material have also been proposed in the context of forging. [Mai+14] is a survey of nondestructive methods for elasticity and plasticity parameter estimation of metal structures and anisotropic foils made of diverse material layers. **FEM** can be used in a preliminary step of sensitivity analyses, which consists in determining the influence of a parameter of interest on measurable quantities. Then, **Proper Orthogonal Decomposition (POD)** often associated with interpolation through **RBF** can be performed to reduce the number of interesting measurable quantities. Actual experiments monitoring only the most interesting quantities according to the previous steps can then be made. Finally, inverse analysis using **FEM** can be performed, seeking to minimize the difference between the experimental measurements with the simulated ones obtained using different sets of parameters. The set of parameters minimizing this difference is assumed to be close to the actual parameters of the object.

Elasticity and plasticity parameter estimation of orthotropic foil-type objects

Engineering has also been interested in parameter estimation. In [GMN12], the authors focus on simulations of cruciform biaxial tests[13] performed using **FEM**. The objects of interest are free standing foils with an orthotropic elastic-plastic-hardening model. Then, the in-plane displacements are measured using **Digital Image Correlation (DIC)**[14], which consists in comparing pictures of the object before and after the test to get the displacements of some specified points of its surface. In this case, the pictures result from simulations. Finally, the authors present two solutions that can be used for the parameter estimation. First, an optimization-based solution using the **TRA** to estimate the parameters is presented. The second solution first uses **POD** to reduce the number of measurable quantities to the ones the most related to the sought parameters. This reduced number of measures are then used to feed an **Artificial Neural Network (ANN)** that estimates the parameters of interest.

Parameter estimation applied to the design of objects with desired behavior

Another parameter estimation developed for engineering purposes is presented in [Bic+10]. The proposed method first establishes a database of base materials deformation properties. This is done automatically by probing the surface of objects using a four-**DOF** robotic arm, a six-axis force-torque sensor and a vision subsystem able to track surface displacements. The base materials are then modeled using a data-driven non-linear stress-strain relation in a **FEM** simulation. The deformation properties are then set such as the simulated deformations match the best the set of observation samples. Finally, the design process is assimilated to an optimization process. This process chooses the best distribution of several possible base materials to match the fabricated object behavior with the input force-displacement examples. The solution was tested by replicating actual objects whose geometry was acquired thanks to a Cyberware scanner. Then, their deformation behavior was acquired using the proposed method, and their replicas were printed using the combination of base materials that best approximates the behavior of the actual objects. Finally, the force-displacement curve of the replicas were acquired and compared with the ones of the actual objects, showing a good similarity.

13. Cruciform biaxial tests consist in applying forces on foils to generate a uniform stress field in its central part.
14. Interested readers may read [HR05] for a more detailed explanation.

Elastic and damping parameter estimation at equilibrium

A data-driven method for deformation capture and modeling of general soft objects has been proposed in [Wan+15]. The proposed method iteratively performs physics-based deformation tracking using low cost depth sensors and physics-based parameter estimation using spacetime optimization.

Figure 2.23 illustrates the workflow of the system. First, the system captures the static shape of the object, i.e. its equilibrium geometry under gravity, and reconstructs a triangular mesh of this object. Then, the dynamic motion induced by an initial deformation is captured in real time using three depth sensors from different points of view. The motion is stored under the form of a time sequence of point clouds. Using the static shape as a template, motion tracking is performed through a physics-based probabilistic method applied on the point clouds. The deformation tracking is formulated as a **Maximum A Posteriori (MAP)** problem and is solved using the **Expectation Maximization (EM)** algorithm. The parameter estimation is done through a spacetime optimization whose goal is to match the simulated trajectories with the tracked one. The material elasticity parameters, dynamic damping coefficients and the reference shape, which can be different from the static shape for soft objects, are estimated alternatively and iteratively through a novel splitting scheme. The system is able to determine different Young moduli when an object is heterogeneous, as shown in Figure 2.24.

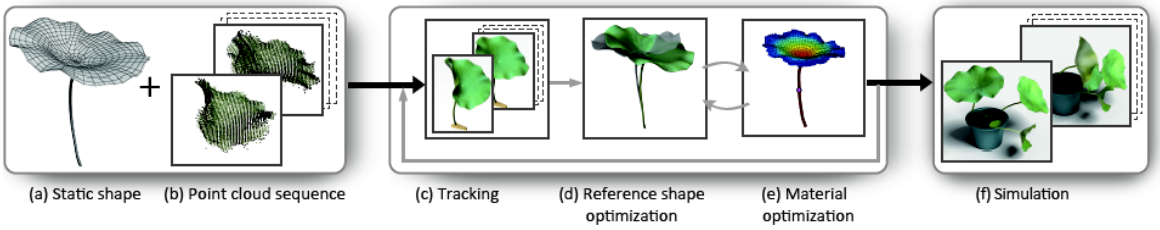


Figure 2.23: Schematic of the framework workflow, from [Wan+15].

However, this method suffers some drawbacks. For instance, it requires an accurate mesh of the object and the high frequency vibrations of the objects cannot be captured using depth sensors working at 30Hz. Finally, the coarse volumetric mesh employed in the **FEM** simulation introduces artificial stiffness.

When grasping the deformable object is intended, additional physical parameters such as the coefficient of friction might come handy.

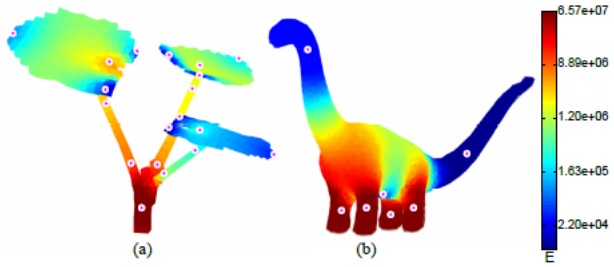


Figure 2.24: Results of the estimation of the Young moduli of an heterogeneous object, from [Wan+15].

Friction coefficient estimation using solely a structured-light range sensor

Another data-based method for an automated system that builds models of heterogeneous non-linear elastic deformable objects has been proposed in [Fon09]. The geometry is acquired thanks to a structured-light range sensor. The coefficient of friction of the object is estimated by performing active probing. Finally, the method permits to render the model in an interactive virtual environment.

The system uses a structured-light range sensor for the geometry capture. It collects data from different orientations and merges them into one model. The dense depth map is recovered from a single image thanks to the structured-light range sensor. The object segmentation is done using an active contour technique. The phase variance, which is proportional to the depth, of the pixels of the image is observed. When two adjacent pixels suffer a huge phase variance, it is interpreted as a boundary between two regions.

For the behavior capture, the force field function is sampled by probing the object at a collection of approximately uniformly distributed surface contact points. Forces of different intensities are repeatedly applied and the resulting displacements are recorded. The probe is a PHANToM haptic device controlled using an open-loop force controller. This probe is not equipped with force or slip sensors. The contact detection is done by detecting acceleration decrease while a constant force is applied.

For the determination of the maximum tangential force component that does not induce slipping of the probe, a binary search strategy is applied, as shown in Figure 2.25. It consists in fixing the normal force and applying different intensities of tangential force to divide each time the interval of admissible tangential force values.

Finally, the system generates a friction model of the object that can be used in a simulation working up to 1,000Hz if a haptic feedback is wanted, or to initialize a parametric method.

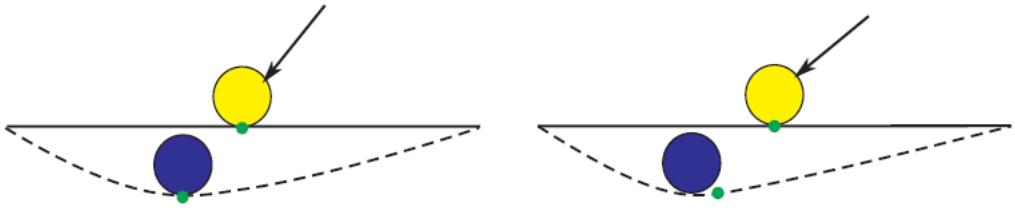


Figure 2.25: Binary search strategy of the friction coefficient, from [Fon09]. The circles represent the probe position and the green points the position of the contact point on the undeformed and deformed shapes. The left picture shows when the probe does not slip, while the right one shows when the probe slips on the surface.

However, the data-acquisition requires several hours.

Semi-automatic registration between medical images and inverse FEM simulation for Young modulus estimation

Finally, parameter estimation may be handful for medical applications. For instance, a method that uses a non-linear real-time **FEM** to perform semi-automatic deformable registration on anatomical structures has been proposed in [Coe+15]. It relies on a constraint-based framework and interactive inverse simulations. The registration is performed for **Computed Tomography (CT)** scans used in the case of radiotherapy treatments. A **CT** scan consists in obtaining cross-sectional images of the body using X-ray.

The registration is done between the **CT** scan and an underlying Finite Element Model. This model is non-linear and uses the formulation of the static Saint-Venant Kirchhoff model[15]. It takes into account non-linear geometrical deformations and integrates over the structure a Hookean constitutive law. The registration method relies on a few control points defined by the user on the first image. Thanks to that, the **FEM** method is able to estimate the external loads, pressures, displacements (thereafter designated by *boundary conditions*). This **FEM** method permits to estimate the Young modulus from deformation observations if the applied forces are known.

Then, a real-time optimization adapts the boundary conditions, such as external loads or pressures, and/or some **FEM** parameters, such as the elasticity parameters of the target, to get the adequate geometrical deformations. The optimization is performed using Lagrange

15. The Saint-Venant Kirchhoff model is an extension of the linear elastic model to the non-linear regime. One interested in hyperelastic modeling may read [GC15].

multipliers to find the static equilibrium of the object, i.e. when the sum of internal and external forces equals to zero.

Finally, the method was tested on seven patient datasets containing each six **CT** scans captured for monitoring their radiotherapy treatment. A comparison between manual segmentation of the parotids and the use of this semi-automatic registration method, both performed by radiologists, was performed using the DICE coefficient[16]. The results have shown a good similarity between the two registration approaches while the semi-automatic is less manpower-consuming.

2.5.3 Conclusion

In this section, we have presented different methods to estimate some physical parameters of unknown objects. First, when proposing such a method, it is necessary to wonder to which purpose will be used the physical parameters. Indeed, it will dictate which assumptions can be made about the object and thus how complex the model must be. For instance, if the object will be subject to small deformations only, a linear elastic model can be sufficient.

2.6 Positioning of this thesis with regards to the literature

This thesis focuses on proposing novel methods to permit the automatic robotic dexterous manipulation of non-rigid objects. We decided to take inspiration from visual servoing, since visual sensors are well-spread in the industry, cheap and non-invasive, to derive a *model-free* deformation servoing method. An advantage of proposing a model-free deformation servoing method is that it does not require heavy preliminary steps. Compared to model-based methods, which require to determine beforehand the physical properties of the object, and to data-based methods, which require to gather a lot of data beforehand, our approach is cheaper in term of time and man-power. Additionally, proposing a model-free method facilitate the transposition from a situation to another, which is not the case when using model-based methods, since the latter require to model the object and its interactions with its environment. Finally, as presented in Table 2.1, most of the methods that permit an overall shape deformation control rely on a physical model or permit to control only a 2D shape. On the contrary, our method permits to control the 3D shape of non-rigid objects.

16. The DICE coefficient is a statistic of similarity between two samples.

Additionally, we decided to propose an interactive elasticity parameter estimation method while most of the methods existing in the literature are offline. Another important difference between the proposed method and the ones from the literature is that it permits to perform simultaneously deformation tracking and elasticity parameter estimation. Our method works in a closed-loop manner, the elasticity parameter estimation using the deformation tracking results and then updating the deformation tracking with the estimated parameters. This closed-loop system permits to remotely estimate deformation contact forces applied on the object using only visual information once the elasticity parameter estimation has converged.

Chapter 3

Active Deformation Control through Visual Servoing of Soft Objects

Contents

3.1 Introduction

In this chapter, we address the problem of active deformation of soft objects. To do so, we propose the ADVISEd (Active Deformation through VIsual SErvoing) method, a novel model-free deformation servoing method able to do an online estimation of the deformation Jacobian that relates the motion of the robot end-effector to the deformation behavior of the object. This approach has been published in [LKM20a].

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.

This chapter first explains the principles of the ADVISEd method. Then, some relevant active deformation methods from the literature are presented. Finally, the ADVISEd method is validated through comparison with the presented state-of-the-art methods. A video illustrating these experiments can be found at this link: https://youtu.be/iovQUcyyuwc.

3.2 Methodology of the ADVISEd method

3.2.1 Generalities

Definition of the term "deformation feature"

In this manuscript, a *deformation feature* is an attribute that can be used to describe the evolution of the deformation of a non-rigid object. In this work, we propose to use a visual-servoing-inspired method. Consequently, we focus on *visual deformation features*, that is simply designated by *deformation features* for the sake of brevity. Some examples of deformation feature are the angle of curvature between some points of the object or the 3D coordinates of some feature points. In this chapter, we consider only the 3D coordinates of some features points, obtained thanks to a depth camera. However, in Chapter 4, we use a different type of deformation feature to allow an easier control of linear deformable objects.

Notation

The formulation that we use is the following:

- k denote the current discrete time. It is only specified when confusion may arise in a formula,

- $\mathbf{v^{ctrl}} \in \mathbb{R}^m$ is the computed end-effector velocity control law vector expressed in the robot end-effector frame, with m the number of **DOFs** of the end-effector(s),

- $\mathbf{e} = \mathbf{s} - \mathbf{s}^* \in \mathbb{R}^n$ 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,

- $\hat{\mathbf{J}}^+ = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T$ 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 $\mathbf{v^e}$ and is defined as:

$$\dot{\mathbf{s}} = \mathbf{J}\mathbf{v^e} \tag{3.1}$$

Open-loop estimation of the deformation Jacobian

A potential open-loop method to estimate the deformation Jacobian is to stimulate successively and individually each of the end-effector(s) **DOFs** while recording the observed deformation. Let j be one of the end-effector(s) **DOF**. An estimate of the coefficient of the deformation Jacobian matrix that relates the i^{th} deformation feature to the j^{th} end-effector(s) **DOF** can be computed as follows:

$$\hat{\mathrm{J}}_{i,j} = \frac{\Delta \mathrm{s}_i}{\mathrm{v}_j \Delta t} \text{for } i \in \{0, \ldots, n-1\},\ j \in \{0, \ldots, m-1\} \quad \text{with } \Delta \mathrm{s}_i = \mathrm{s}_i(k) - \mathrm{s}_i(k-1) \in \mathbb{R} \tag{3.2}$$

where v_j represents a constant velocity input that is applied successively along (for translations) or around (for rotations) each axis of the effector frame during a period Δt and the subscript i denotes the i^{th} row of a vector or a matrix.

However, such a simple method is highly sensitive to noise and is not efficient with deformable objects. Indeed, in most cases non-rigid objects have a non-linear and/or unhomogeneous deformation behavior. Consequently, the deformation Jacobian is only valid in a small region around the initial contact between the end-effector(s) and the non-rigid object, and for small deformations only. To avoid this problem, an adaptive method that online estimates the deformation Jacobian is required.

3.2.2 ADVISEd method

We aim at proposing a novel adaptive model-free deformation servoing method to control the 3D shape of an object. Our novel model-free deformation servoing method online estimates the deformation Jacobian using a weighted least-squares minimization method with a sliding window. The deformation Jacobian is updated only with regard to the **DOFs** for which a confidence criterion has been met. This confidence criterion allows to filter noise measurement when the system converges towards the desired deformation.

Online estimation of the deformation Jacobian

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:

$$\dot{\mathrm{s}}_i = \mathbf{v^e}^T \hat{\mathbf{J}}_{i,:}^T + \nu_i \in \mathbb{R} \text{ for } i \in \{0, ..., n-1\} \tag{3.3}$$

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

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

$$C\left(\hat{\mathbf{J}}_{i,:}^T\right) = \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} \left(\dot{\mathrm{s}}_i(j) - \mathbf{v^e}^T(j-1)\hat{\mathbf{J}}_{i,:}^T \right)^2 \right) \tag{3.4}$$

where N is the size of a sliding window, the index j indicates that the sum is performed over previous discrete times, $0 < \Upsilon \leq 1$ is a constant forgetting factor giving less importance to the oldest measures and $m(k) = \sqrt{1 + \mathbf{v^e}^T(k)\mathbf{v^e}(k)}$ is a normalization signal. Let's define:

$$\mathbf{R} = \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} \mathbf{v^e}(j-1)\mathbf{v^e}^T(j-1) \right) \in \mathbb{R}^{m,m} \tag{3.5}$$

$$\mathbf{Q} = \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} \mathbf{v^e}(j-1)\dot{\mathrm{s}}_i(j) \right) \in \mathbb{R}^{m,1} \tag{3.6}$$

The cost function can be rewritten as follows:

$$C = \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} \dot{\mathrm{s}}_i^2(j) \right) + \hat{\mathbf{J}}_{i,:} \left(\mathbf{R} \hat{\mathbf{J}}_{i,:}^T - 2\mathbf{Q} \right) \tag{3.7}$$

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 $\bigtriangledown C$ with regards to $\hat{\mathbf{J}}_{i,:}^T$, where $\bigtriangledown C = -2\mathbf{Q} + 2\mathbf{R}\hat{\mathbf{J}}_{i,:}^T$.

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

$$\hat{\mathbf{J}}_{i,:}^T = \mathbf{R}^{-1}\mathbf{Q} \in \mathbb{R}^{m,1} \tag{3.8}$$

Adaptation of the update of the deformation Jacobian

In practice, it is not possible to directly implement Equation (3.8). Indeed, if the system is not well stimulated, which is notably the case when the 3D shape of the object is becoming close to the desired one , 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, inspired from the work of De Mathelin and Lozano presented in [DL99], is used to determine how the deformation Jacobian will be updated. From Equation (3.5), we can see that the matrix $\mathbf{R}$ is a positive-definite symmetric $m \times m$ matrix.

Proof that R is positive-definite: A matrix is positive-definite if $\forall \mathbf{z} \neq \mathbf{0}_m \in \mathbb{R}^m$, with $\mathbf{0}_m$ the null vector, we have:

$$\mathbf{z}^\top \mathbf{R} \mathbf{z} > 0 \tag{3.9}$$

From Equation (3.5), Equation (3.9) becomes:

$$\begin{aligned}
\mathbf{z}^\top \mathbf{R} \mathbf{z} &= \mathbf{z}^\top \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} \mathbf{v}^{\mathbf{e}}(j-1) \mathbf{v}^{\mathbf{e}T}(j-1) \right) \mathbf{z} \\
&= \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} \mathbf{z}^\top \mathbf{v}^{\mathbf{e}}(j-1) \mathbf{v}^{\mathbf{e}T}(j-1) \mathbf{z} \right) \\
&= \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} (\mathbf{v}^{\mathbf{e}}(j-1)^\top \mathbf{z})^\top (\mathbf{v}^{\mathbf{e}T}(j-1) \mathbf{z}) \right) \\
&= \sum_{j=k-N}^{k} \left(\frac{\Upsilon^{k-j}}{m^2(j-1)} \| \mathbf{v}^{\mathbf{e}T}(j-1) \mathbf{z} \|^2 \right) > 0
\end{aligned} \tag{3.10}$$

Thus, Equation (3.10) proves that $\mathbf{R}$ is positive-definite, since Υ and m^2 are positive scalars

and $\|\mathbf{v}^{\mathbf{e}T}(j-1)\mathbf{z}\|^2$ is positive.

Consequently, the matrix $\mathbf{R}$ can be decomposed such as:

$$\begin{aligned} \mathbf{R} &= \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi} \\ \text{with} \quad & \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi_0} & \dots & \boldsymbol{\phi_{m-1}} \end{bmatrix} \in \mathbb{R}^{m,m} \\ \text{and} \quad & \boldsymbol{\Gamma} = \begin{bmatrix} \gamma_0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \gamma_{m-1} \end{bmatrix} \in \mathbb{R}^{m,m} \end{aligned} \tag{3.11}$$

where $\gamma_0 \geq \cdots \geq \gamma_{m-1} \geq 0$ are the eigenvalues of $\mathbf{R}$ and $\boldsymbol{\phi_0}, \dots, \boldsymbol{\phi_{m-1}}$ are its eigenvectors.

The update of a row of the estimated deformation Jacobian matrix is performed depending on a user-defined confidence threshold ϵ:

$$\begin{aligned} \hat{\mathbf{J}}^T_{i,:}(k) = & \begin{cases} \mathbf{R}^{-1}\mathbf{Q} \text{ if } \gamma_0 > \epsilon, \ \dots, \ \gamma_{m-1} > \epsilon \\ \hat{\mathbf{J}}^T_{i,:}(k-1) \text{ if } \gamma_0 \leq \epsilon, \ \dots, \ \gamma_{m-1} \leq \epsilon \\ \mathbf{V_1}\mathbf{Q} + \mathbf{V_2}\hat{\mathbf{J}}^T_{i,:}(k-1) \text{ otherwise} \end{cases} \\ \text{with} \quad & \mathbf{V_1} = \boldsymbol{\Gamma}_{:,0:j}\boldsymbol{\Phi}^{-1}_{0:j,0:j}\boldsymbol{\Gamma}^T_{:,0:j} \\ \text{and} \quad & \mathbf{V_2} = \boldsymbol{\Gamma}_{:,j+1:m-1}\boldsymbol{\Gamma}^T_{:,j+1:m-1} \\ & \text{with } \phi \text{ sorted such as } \phi_j > \epsilon \text{ and } \phi_{j+1} \leq \epsilon \end{aligned} \tag{3.12}$$

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 Equation (3.12) 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.

Finally, the global estimated deformation Jacobian matrix $\hat{\mathbf{J}}$ is obtained by updating each of its rows individually using the update method presented above.

3.2.3 Marker-based method

The deformation of the object is tracked using η passive visual markers, called feature points thereafter, attached to the surface of the object. The deformation vector groups the 3D position of each marker with regard to the camera and can directly be retrieved from the depth sensor data. The choice of the number of feature points used to construct the deformation vector can

be defined by the user. The deformation vector can be written as follows:

$$\mathbf{s}(k) = \Big(x_0(k)\ y_0(k)\ z_0(k)\ \ldots\ x_{\eta-1}(k)\ y_{\eta-1}(k)\ z_{\eta-1}(k)\Big)^T \in \mathbb{R}^n \tag{3.13}$$

In our work, we consider an eye-to-hand configuration. Consequently, the velocity control law expressed in the end-effector frame that is used in the marker-based method is as follows:

$$\mathbf{v}^{\mathbf{ctrl}} = -\alpha \hat{\mathbf{J}}^{+}\mathbf{e} \in \mathbb{R}^m \tag{3.14}$$

where α is the control law gain and $\hat{\mathbf{J}}^{+}$ is the Moore-Penrose pseudo-inverse of the deformation Jacobian that is estimated online and relates the variations of the deformation features expressed in the camera frame to the variations of the motions of the end-effector(s) expressed in the end-effector frame.

Note: in case of an eye-in-hand configuration, the control law would be computed in the camera frame and equal to:

$${}^{c}\mathbf{v}^{\mathbf{ctrl}} = -\alpha \hat{\mathbf{J}}_c{}^{+}\mathbf{e} \in \mathbb{R}^m \tag{3.15}$$

where $\hat{\mathbf{J}}_c$ is an estimation of the deformation Jacobian expressed in the camera frame. Because in practice we apply the control law in the end-effector, the frame in which the velocity control is expressed would be changed as follows:

$$\mathbf{v}^{\mathbf{ctrl}} = {}^{e}\mathbf{V}_c\, {}^{c}\mathbf{v}^{\mathbf{ctrl}} \tag{3.16}$$

where ${}^{e}\mathbf{V}_c$ is the spatial motion transform matrix that transforms velocities expressed in the camera frame into the effector frame and is expressed as:

$${}^{e}\mathbf{V}_c = \begin{bmatrix} {}^{e}\mathbf{R}_c & [{}^{e}\mathbf{t}_c]_X\, {}^{e}\mathbf{R}_c \\ \mathbf{0}_{3\times3} & {}^{e}\mathbf{R}_c \end{bmatrix} \tag{3.17}$$

where ${}^{e}\mathbf{R}_c$ is the rotation matrix expressing the rotation between the camera frame and the effector frame, $[{}^{e}\mathbf{t}_c]_X$ is the skew symmetric matrix associated to the translation vector expressing the translation between the origin of the two frames and $\mathbf{0}_{3\times3}$ is a 3-by-3 null matrix.

3.2.4 Marker-less method

A 3D point cloud, captured by a remote **RGB-D** camera, composed of η 3D points located in a given **Surface of Interest (SoI)** is used as deformation feature vector $\mathbf{s}(k) =$

$\Big(x_0(k)\ y_0(k)\ z_0(k)\ \ldots\ x_{\eta-1}(k)\ y_{\eta-1}(k)\ z_{\eta-1}(k)\Big)^T$. To ensure that the points of the point cloud correspond to the same physical points of the non-rigid objects, the following assumptions are made:

- the velocity of the non-rigid object with the regard to the camera is null, i.e. the object does not have a rigid motion relatively to the camera,
- the deformations that the non-rigid object undergoes are small enough not to generate high lateral motions relatively to the remote camera.

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 [Hub81]. We apply Iteratively Reweighted Least-Squares (IRLS) method in order to convert the M-estimation problem into an equivalent weighted least-squares problem, as proposed in [PM06]. The error used in the computation of the control law becomes $\mathbf{e}'(k) = \mathbf{W}\Big(\mathbf{s}(k) - \mathbf{s}^*\Big) \in \mathbb{R}^n$, 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 [CH80].

The use of M-estimation turns the control law Equation (3.15) used for the marker-based method into:

$$\mathbf{v}^{\mathbf{ctrl}}(k) = -\alpha\Big(\mathbf{W}\hat{\mathbf{J}}(k)\Big)^+\mathbf{e}'(k) \in \mathbb{R}^m \tag{3.18}$$

3.3 State-of-the-art methods used as references

We decided to compare our method with two active deformation methods: a model-based method and a model-free method. This section first presents the model-based one and then the model-free one.

3.3.1 Model-based deformation method

Several model-based deformation servoing methods have been proposed in the literature (see Section 2.4). The one used for the evaluation of the ADVISEd method is similar to the one proposed by Adagolodjo *et al.* in [Ada+16]. This model-based method estimates the deformation Jacobian using physics-based corotational FEM simulation. We implemented this method using the SOFA framework [Fau+07]. The velocity control law is given by $\mathbf{v}^{\mathbf{ctrl}}(k) = -\alpha\hat{\mathbf{J}}^+\mathbf{e}(k) \in \mathbb{R}^m$

where m is the number of **DOFs** of the end-effector and α is the control law gain. Several physics-based simulations are run in order 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. The partial deformation Jacobian matrices $\hat{\mathbf{J}}^{dir_j} \in \mathbb{R}^{n,1}$ are estimated such as

$$\hat{\mathrm{J}}_i^{dir_j} = \frac{\Delta \mathrm{s}_i}{\mathrm{v}_j \Delta t},\ i \in \{0, ..., n-1\},\ j \in \{0, ..., m-1\} \tag{3.19}$$

where v_j is the constant input velocity of the simulated end-effector in the j^{th} direction, the subscripts i and j indicate the indices of an element in a vector, Δt corresponds to the duration between the previous and current estimation and $\Delta \mathrm{s}_i = \mathrm{s}_i(k) - \mathrm{s}_i(k-1)$.

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

3.3.2 Model-free deformation servoing

We also compare our approach with the model-free deformation servoing method proposed by Navarro-Alarcón *et al.* in [Nav+13]. This method estimates the deformation Jacobian matrix $\hat{\mathbf{J}}$ from the deformation feature vector and the robot velocity observation using the Broyden iterative update rule [Bro65]:

$$\hat{\mathbf{J}}(k) = \hat{\mathbf{J}}(k-1) + \varepsilon \frac{\Delta \mathbf{s}(k) - \hat{\mathbf{J}}(k-1)\Delta \mathbf{x}(k)}{\Delta \mathbf{x}^T(k)\Delta \mathbf{x}(k)} \Delta \mathbf{x}^T(k) \in \mathbb{R}^{n,m} \tag{3.20}$$

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. Consequently, 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 resulting velocity control law is:

$$\begin{aligned} \mathbf{v}^{\mathbf{ctrl}}(k) &= \alpha \hat{\mathbf{J}}^{+}(k) \Big(\mathbf{p}(k) - c\ sat\Big(K_s \mathbf{e}(k)\Big) \Big) \in \mathbb{R}^m \\ \text{with } \dot{\mathbf{p}}(k) &= -sat\Big(K_s \mathbf{e}(k)\Big) - \mathbf{C}\mathbf{p}(k) \in \mathbb{R}^n \end{aligned} \tag{3.21}$$

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

3.4 Evaluation during marker-based tracking experiments

Several marker-based experiments were conducted to evaluate the different methods. First, stand-alone experiments were conducted to evaluate the accuracy of the ADVISEd method according to the number of markers and the robustness of the method against external perturbations . A second type of experiments aims to compare the accuracy and stability of the ADVISEd method with these of state-of-the-art methods.

3.4.1 Initialization of the Jacobian

The deformation Jacobian of both the ADVISEd and the model-free state-of-the-art deformation methods is initialized using the open-loop method presented in Section 3.2.1. This offline initialization takes a few seconds.

3.4.2 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 Figure 3.1 for the setup). A sheet of foam, shown in an inset of Figure 3.1, is deformed in order to reach a desired curved shape. A Intel Realsense D435 **RGB-D** remote camera is used for tracking both the robot and the soft object. The depth information is accurate within to $\pm$0.5mm. Since the depth sensor is a bit noisy for accurate measurements, the depth data is filtered using an Infinite Impulse Response (IIR) Butterworth low-pass filter with a cutoff frequency of 2Hz.

Tracking of the deformation

As mentioned in Section 3.2.3, for the marker-based active shaping task, the deformation of the object is tracked using η passive visual markers attached to the object surface. These markers, called feature points thereafter, are 38-millimeter white spheres and are tracked in the RGB image using ViSP library [MSC05]. The tracking method is based on a binarisation of a grayscale version of the RGB image. Once the binarisation done, a contour detection method using the Freeman chain coding is used to determine some characteristics of a group of

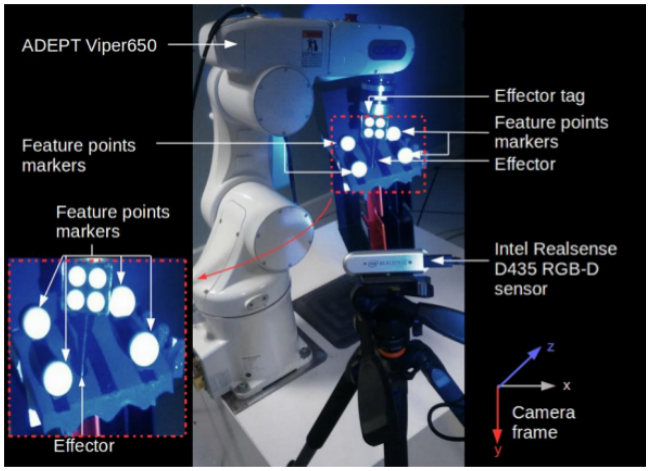


Figure 3.1: Experimental setup for the marker-based experiments.

connected pixels. By comparing the characteristics of the detected groups of connected pixels with the last known characteristics of the connected pixels that form a marker, it is possible to track a marker over time.

3.4.3 Experimental Evaluations

For the marker-based active shaping task, the considered error function is the Euclidean norm of the deformation error vector. For instance, the initial error is denoted $e_0 = \|\mathbf{e}(0)\|$.

The evaluation of performances takes into account several criteria, illustrated in Figure 3.2:

- t_r: response time of the system, expressed in seconds, which corresponds to the time needed to reduce the error function to 10% of the initial error 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 $[e_\infty - 0.05e_0 \ ; \ e_\infty + 0.05e_0]$,
- $e_\infty = \|\mathbf{e}(k \to \infty)\|$: 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

Stand-alone evaluation

We conducted experiments to determine the impact of the number of feature points on the task completion, using at most three feature points and a control law gain of 0.05. In the case of manipulation of rigid objects by rigid end-effector(s), the use of one feature point leads to an under-actuated system while the use of three feature points leads to an over-actuated system. We thus decided to evaluate the method performances using up to three feature points, as the use of a greater number of feature points leads to very high occlusion and often leads to tracking loss. Dealing with occlusion and tracking loss goes beyond the scope of this work. The results are summarized in Table 3.1. The results show that the system is faster when using only one feature point. However, the system being under-constrained, the results are more variable between the manipulations: even if the system is able to converge, the final offset differs greatly when changing the control law gain. When using three feature points, the system converges with a higher final offset. This is because in this case the system is over-constrained and can be stuck in a local minimum steady state. These results motivated us to perform the other evaluations using only two feature points.

Table 3.1: Comparison of the results according to the number of feature points

Number of markers	**Gain**	t_r **(s)**	t_s **(s)**	e_∞ **(mm)**	d_{eff} **(mm)**
1	0.05	20	230	0.2	16.8
2	0.05	70	125	0.7	16.3
3	0.05	75	400	2.5	19.6
1	0.025	35	430	1.3	20.5
2	0.025	220	300	1	20.9
3	0.025	140	145	1.8	17.2

To determine the adaptability of our method to external changes, we performed experiments where the system had to reach several successive targets. Figure 3.3 illustrates the results of one of these experiments. It was conducted with a control law of 0.025, controlling two feature points in order to reach three successive targets. The first and second tasks were designed such as the robot should exert a pressure on the object to reach its targets. The third task was designed such as it should release the pressure. The plot shows that the system was able to make all the **DOFs** of the different feature points converge during each task.

Comparison with the state-of-the-art methods

Figure 3.4 illustrates the evolution of the Euclidean norm of the deformation error vector obtained with the different methods during the execution of 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 vector and deformation error vector was n= 6. Each algorithm was optimally tuned to achieve the best trade-off between accuracy, stability and responsiveness. For ADVISEd, the parameters were as follows: Υ= 0.99, N= 10, ϵ = 0.001, α = 0.05. For the model-based method, the parameters were as follows: Young modulus = 13kPa, Poisson's ratio = 0.3, Rayleigh stiffness = 0.125, mass density = $40kg.m^{-3}$. For the model-free method, the parameters were as follows: $\mathbf{C} = 0.5\mathbf{I}$, $c = 1$, saturation threshold = 0.02, $K_s = 3$, $\varepsilon = 0.3$, $\alpha = 0.0125$. The results of these experiments are summarized in Table 3.2.

Table 3.2: Comparison of the performances of the different algorithms performing a marker-based active shaping task.

Method	t_r **(s)**	t_s **(s)**	e_∞ **(mm)**	d_{eff} **(mm)**
ADVISEd	70	125	0.7	16.3
model-based	40	1130	3.1	63.5
model-free	35	175	0.8	16.3

The model-based method, whose results are depicted in blue in Figure 3.4, was less stable (d_{eff} = 63.5mm compared to 16.3mm for the two model-free methods) and less accurate (e_∞ = 3.5mm) 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 of accurately obtaining the physical parameters necessary for modeling the geometry and physical properties of a deformable object as well as those of the surrounding scene. This reinforced our idea of proposing a model-free method to avoid modeling constraints. The ADVISEd method and the 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 = 35s compared to 70s) but needed more time to become stable than the ADVISEd method (t_s = 175s compared to 125s). We thus decided to conduct additional experiments to compare these two methods.

3.5 Evaluation during marker-less tracking experiments

3.5.1 Experimental Setup

We used the same hardware as presented in Section 3.4.2. Several soft objects were manipulated: a flat piece of foam, a soft ball and a soft toy. The experimental setup is shown in Figure 3.5, the soft objects being presented as insets.

For this series of experiments, we used the evaluation method presented in Section 3.2.4. The projected area of the **SoI** on the surface of the object was set to $3cm \times 3cm$. Different downsampling 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 ensure a fair comparison between the ADVISEd method and the model-free method proposed in [Nav+13], the velocity control law shown in Section 3.3.2 has been adapted in order to also use M-estimation. The resulting law is given by:

$$\mathbf{v}^{\mathbf{ctrl}}(k) = \alpha\Big(\mathbf{W}\hat{\mathbf{J}}(k)\Big)^{+}\mathbf{W}\Big(\mathbf{p}(k) - c\ sat\Big(K_s\mathbf{e}(k)\Big)\Big) \in \mathbb{R}^m \tag{3.22}$$

Let $e_H(k)$ be the Hausdorff distance between the desired and observed surface at the discrete time k [ASE02]. For the different marker-less tasks, the final error that is considered is $e_{H\infty} = e_H(k \to \infty)$.

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 (Figure 3.6a to Figure 3.6c),
- c) bringing back a deformed shape of the soft toy to a desired less deformed shape (Figure 3.7a to Figure 3.7c),
- 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 regard to external disturbances.

3.5.2 Comparison of the results

During each of these experiments, the state-of-the-art method parameters used were as follows: $\mathbf{C} = 0.5\mathbf{I}^{6\times6}$, $c = 1$, saturation threshold $= 0.02$, $K_s = 3$, $\varepsilon = 0.3$ and $\alpha = 0.045$.

For experiments a) b) and c), several downsampling of the point cloud located in the **SoI** were tested to determine the impact of the number of points η on task completion. During these experiments, the ADVISEd method parameters were as follows: $\alpha = 0.045$, $\Upsilon = 0.99$, $N = 10$ and $\epsilon = 0.001$. The results of these experiments are presented in Table 3.3.

Table 3.3: 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	4×4	ADVISEd	30	80	2.1	21.3
a	4×4	model-free	20	50	2.1	26.2
a	5×5	ADVISEd	40	90	1.0	35.0
a	5×5	model-free	5	90	2.1	21.1
b	6×6	ADVISEd	40	90	2.0	28.2
b	6×6	model-free	20	90	2.0	23.4
b	12×6	ADVISEd	150	160	1.0	16.4
b	12×6	model-free	5	50	1.0	23.5
c	5×5	ADVISEd	90	90	2.1	34.5
c	5×5	model-free	N/A	N/A	div.	N/A
c	6×6	ADVISEd	200	210	1.0	48.0
c	6×6	model-free	10	20	1.0	42.4
c	9×9	ADVISEd	180	220	2.1	131.1
c	9×9	model-free	75	220	3.1	66.5

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 η 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. The results of experiment b) with a **SoI** comprising $\eta = 5 \times 5$ points are depicted in Figure 3.8. 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. The ADVISEd method presents a smoother behavior than the model-free state-of-the-art

method.

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 3.4. Increasing the size of the weighting window and decreasing the value of the forgetting factor both increase the robustness of the method with regard to external perturbations by reducing the impact of occasional outliers. Increasing the value of the confidence threshold increases the robustness of the method by filtering noise measurements.

Table 3.4: Evaluation of the performances of the ADVISEd method when subject to external disturbances. t_s corresponds to the duration between the last disturbance and the steady state.

Υ	N	ϵ	t_r **(s)**	t_s **(s)**	e_∞ **(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

The Hausdorff distance being sensitive to the camera sensor noise, we chose to represent instead the mean square error $e_{ms}(k)$ for display purpose only. Figure 3.9 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: Υ= 0.99, N= 10, $\epsilon = 0.001$ and $\alpha = 0.045$. 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 Figure 3.9. 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 makes them suitable for different kind of applications: the ADVISEd method is well-suited to applications where smooth behavior is required whereas the state-of-the-art method is more suitable when a high responsiveness is desired and overshoot is not critical.

3.6 Conclusion

In this chapter, we presented the ADVISEd method, a model-free deformation servoing method based on a numerical estimation of the deformation Jacobian relating the end-effector motions to the object shape changes. 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. The latter is efficiently updated by determining the reliability of the observations using eigen-value decomposition and a confidence threshold.

Several experiments were conducted in order 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 an accurate knowledge of the physical parameters of the object and its neighborhood to ensure correct results whereas the two model-free methods did not suffer from this issue.

Further marker-less experiments 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 **SoI** as well as external disturbances. Experiments have shown 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 to interactively control the deformations of soft objects.

Due to the tracking method used, the marker-less experiments were performed under the assumptions that the non-rigid object does not move relatively to the remote camera and that its deformations are small enough not to generate high lateral motions relatively to the remote camera. While this thesis is not focused on deformation tracking, improvements to our method could still be made in that direction. For instance, to avoid the aforementioned limitations, additional computer vision algorithms could be used, ensuring a registration between the physical points and the point cloud over time. An example of such an algorithm is the Scale Invariant Feature Tracking method proposed by Lowe in [Low99].

We thus decided to improve the ADVISEd method to allow dual-arm robotic manipulation of linear deformable objects. To do so, we propose a deformation servoing method along with a marker-less tracking method ensuring a registration between the physical points of the object and the deformation features. This work will be presented in the next chapter.

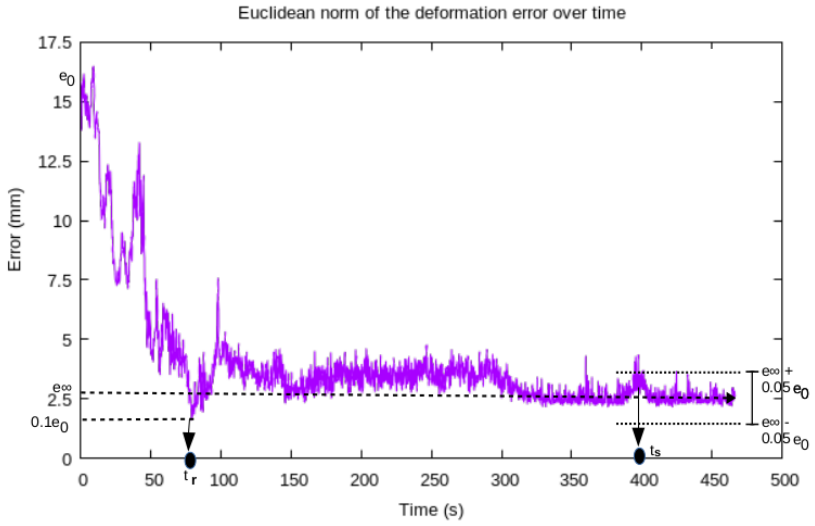


Figure 3.2: Illustration of some of the evaluation criteria for the case of a manipulation with three feature points.

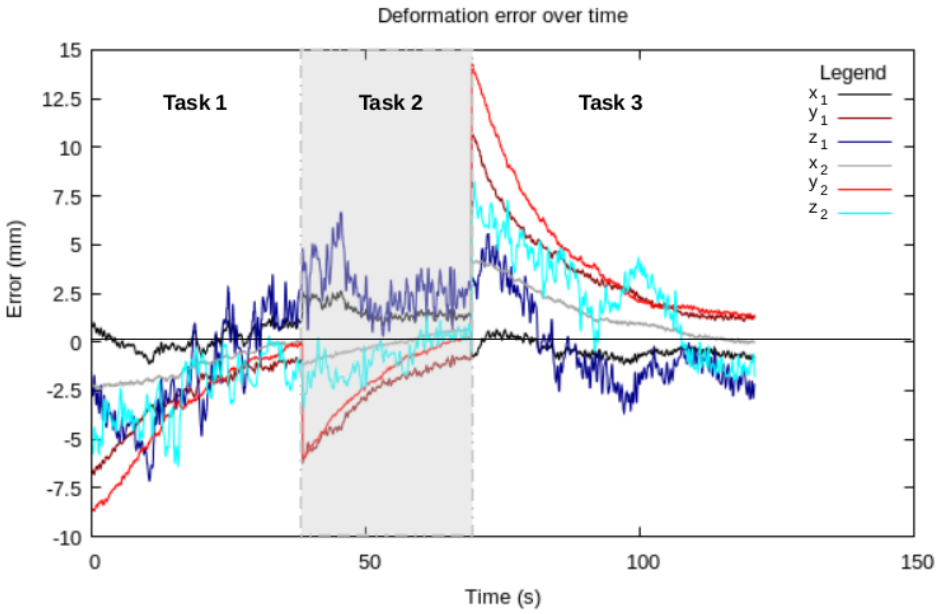


Figure 3.3: Completion of successive tasks, with (x_1, y_1, z_1) and (x_2, y_2, z_2) being the 3D coordinates of the 1^{st} and 2^{nd} marker respectively.

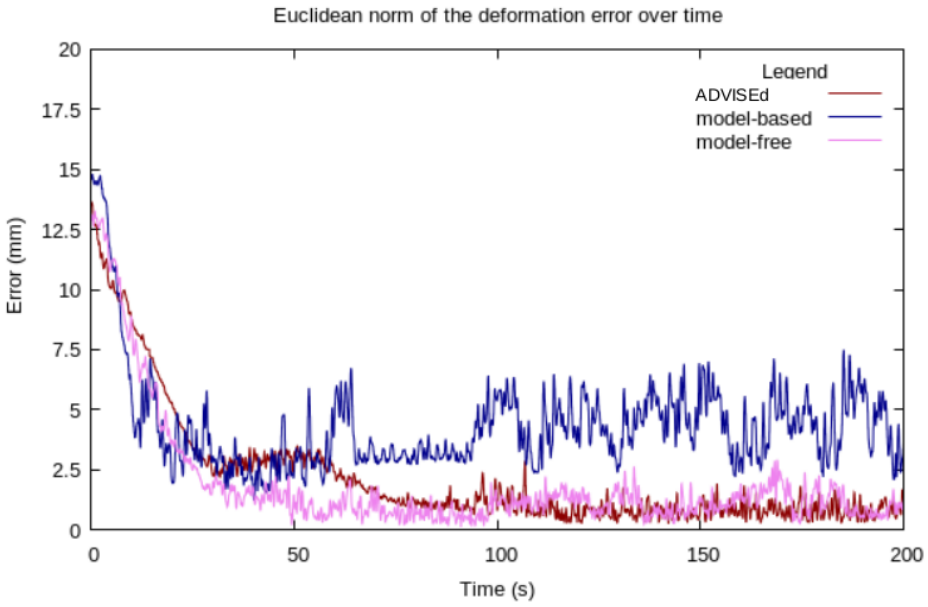


Figure 3.4: Euclidean norm of the deformation error for the three methods during a marker-based active shaping task.

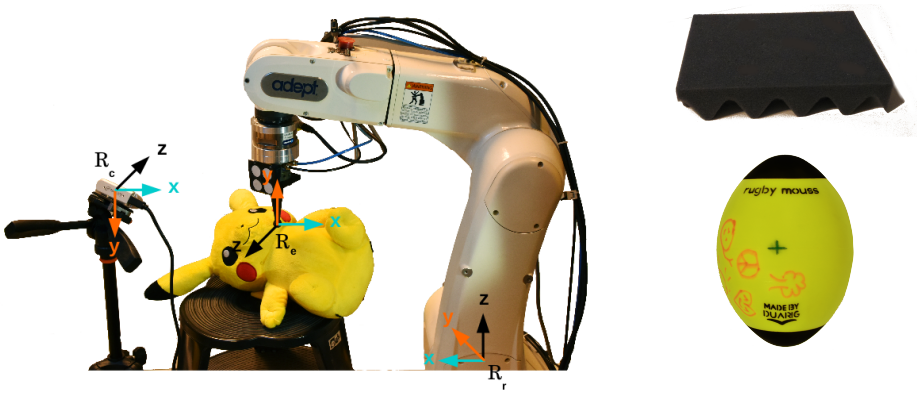
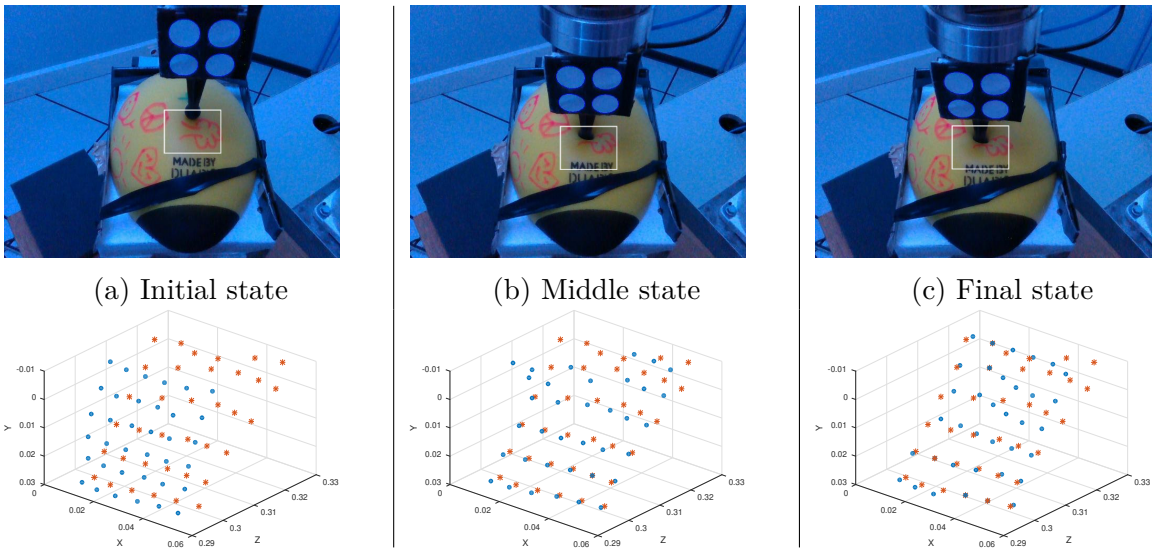
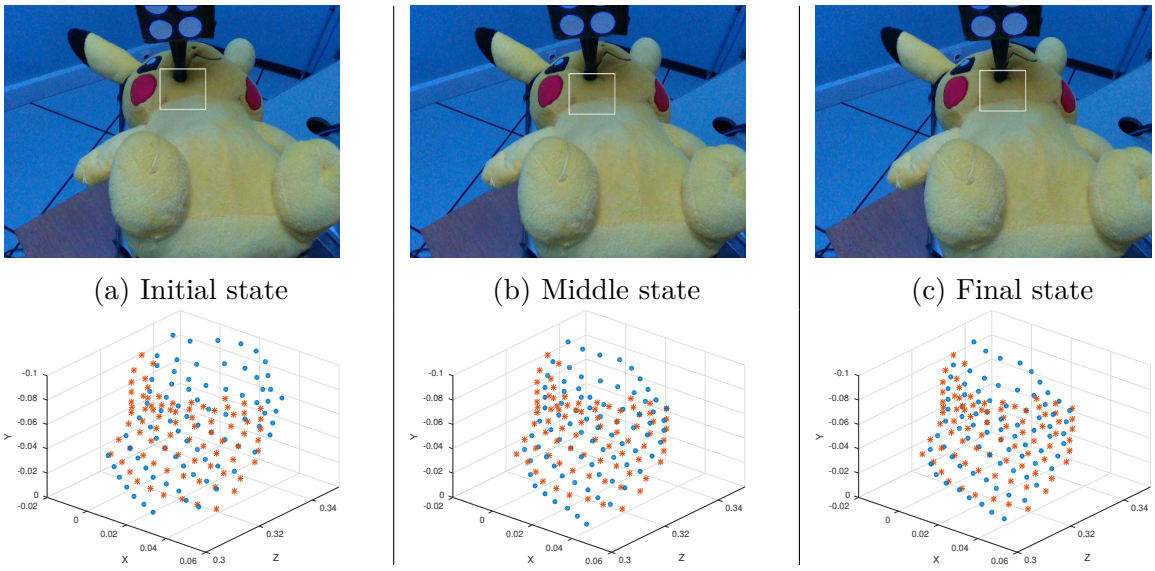


Figure 3.5: Experimental setup showing the **RGB-D** camera and the robot performing a soft object shape preservation task. On the right-hand side are presented the other deformable objects used during the experiments.



(a) Initial state (b) Middle state (c) Final state

Figure 3.6: Examples of active shaping (a to c) tasks. 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 in the figures.



(a) Initial state (b) Middle state (c) Final state

Figure 3.7: Examples of shape preservation tasks(a to c). 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.

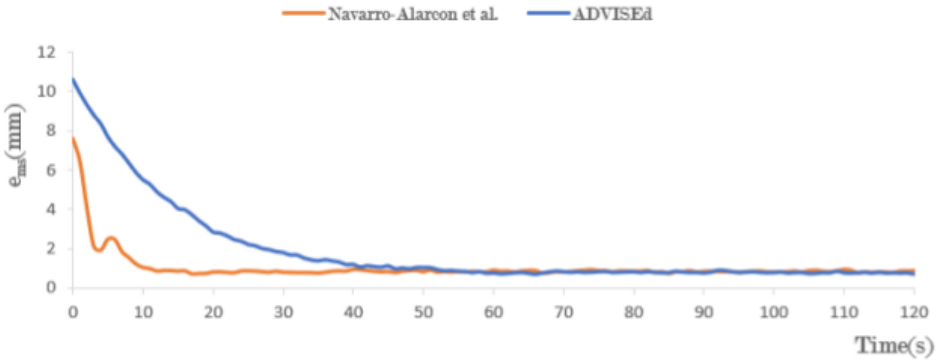


Figure 3.8: Comparison of the evolution of the mean square error function during an active shaping task.

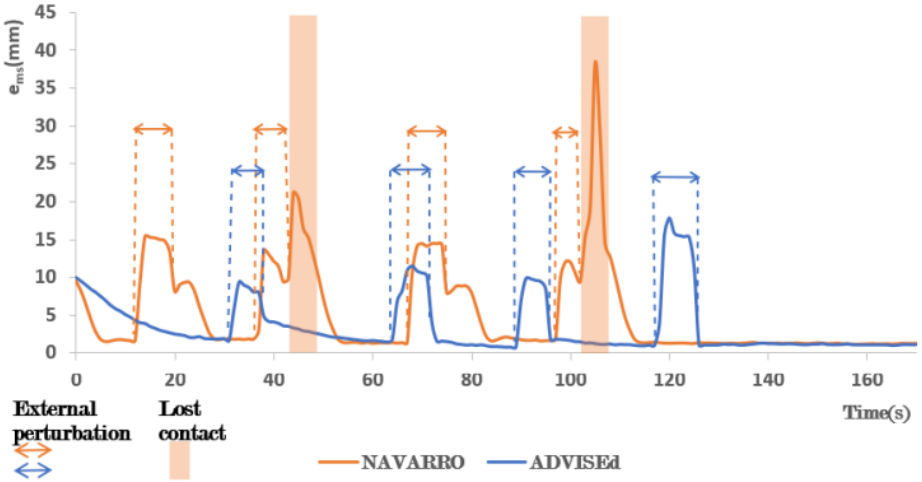


Figure 3.9: Evolution of the mean square error function during an active shaping task when external disturbances occur.

Chapter 4

Automatic Shape Control of Deformable Wires based on Model-Free Visual Servoing

Contents

4.1 Introduction

Deformable wires are used in various fields and the control of their shapes by robots is of interest for many applications, such as in medicine for robotic needle insertion [CKN15] or the control of catheter [Beh+19]; [KH11]; [Fan+19] or in industry for automatic routing of wires [Zhu+19]. However, nowadays, industrial robots still cannot easily perform such tasks due to the complex behavior of wires subject to deformations.

In this chapter, we present a novel approach to automatically control the 3D shape of deformable wires. This approach has been published in [LKM20b]. The automatic control of the shape of an object will be thereafter denoted *shape servoing*. In Chapter 3, we presented a novel method for shape servoing of generic deformable objects. In this chapter, we propose to derive from this work a method that allows dual-arm manipulation of linear deformable objects. Our contributions are as follows:

- a novel kind of visual feature that permits to determine the 3D shape of homogeneous deformable wires using a geometric model;
- a marker-less model-free tracking method based on particle filtering that permits to track at an interactive time a deformable wire and that is independent of the mechanical properties of the wire;
- an adaptive model-free shape servoing method that controls two robot end-effectors to make a wire reach a desired 3D shape by moving solely its extremities.

Our approach does not require any mechanical information on the wire or a marker on it to perform the control of the 3D shape. In addition, computation time performances allow its use at an interactive time to handle various manipulations and transformations, even for large deformations.

The remainder of this chapter is organized as follows: Section 4.2 presents the related work on wire tracking and wire shape control, Section 4.3 describes the proposed methods while Section 4.4 details the experimental evaluation, which is illustrated in the video available here. Finally 4.5 ends the chapter with a discussion on the approach.

4.2 Related work

Controlling the shape of a deformable wire requires to track its 3D shape at an interactive time and to provide a control law to make it converge towards a desired 3D deformed shape.

Therefore, in this section, we first review wire tracking and then wire shape control. This literature review is separated from the literature review of Chapter 2 because it focuses on the particular case of deformable wires.

4.2.1 Wire tracking

In order to control the deformation of a wire, it is necessary to be able to track its shape. Matsuno *et al.* have proposed a method that focuses on a particular case of deformation: the knotted wire [MF06]. The method uses a topological model that relies on knot theory in order to represent a wire. The use of invariants of knots allows to assess whether two knots are similar or not.

Image processing techniques have been used to track a deformable wire in the image plane in [AW00]. In the initialization phase, the user must select a seed point in a region of the image that matches one extremity of the wire. Some characteristics of this region are extracted, such as luminance or global orientation. Then, the method scans the whole image region by region to detect the overall shape of the wire by finding a region with similar characteristics.

More recently, Padoy *et al.* have proposed a wire tracking method based on texture augmentation of the wire [PH12]. The wire texture is augmented by adding a pattern composed of alternating colors. An energy minimization method is used to adapt a NURBS spline model, representing the 3D configuration of the wire, with the visual information captured by calibrated stereo cameras.

Probabilistic methods have been proposed for deformable object tracking. For instance, a probabilistic framework working with physical simulations to track deformable wires using point clouds is proposed in [TT18]. Consequently, knowledge of the physical properties of the object is required. Another method is proposed in [CB19], which consists of a probabilistic registration method that uses visibility information to disregard occluded model points for the registration. An history of previously encountered situations is recorded to perform a k-Nearest Neighbors (k-NN) template recovery when tracking failures occur. Finally, a probabilistic approach that considers a physical model of the object has been proposed in [Sch+13] for tracking deformable objects from point clouds.

4.2.2 Wire shape control

We propose to divide wire shape control methods into two categories, whether or not they use mechanical models to predict the wire shape or its interactions with the environment.

To control the shape of a deformable wire, methods relying on physics-based models have been proposed to predict the behavior of the wire. For instance, Laranjeira *et al.* have proposed a visual servoing method to control the slackness of a tether that connects two robots [LDH17]. This method uses a mechanical model based on the cathenary analytical formulation to represent the tether. The method is based on the assumption that gravity is responsible for the tether deformation.

A dead zone compensation model and a Coulomb friction compensation model have been employed in a combined force-position control method in [KH11]. This method enables a catheter-like robot to be able to make contact with a moving target at a desired force. These two models compensate for backlashes and frictions resulting from the movements in the environment.

Alternatively, mass-spring models have been used to control the deformation of a wire. For instance, the control of the oscillations that occur when a wire is handled at one of its extremities by a gripper has been addressed in [SH02]. The wire is modeled as a 1D mass-spring model. Using a force-torque sensor, it is possible to deduce the oscillations of the wire and consequently to determine the forces to be applied to cancel them. These deduced forces permit to plan the positions that the gripper must reach to compensate the oscillations. The oscillation attenuation challenge has also been addressed using the Finite Element Method (FEM) [Din+12]. A local dynamic model of the wire is proposed. This model is used to derive a position-based control law whose stability is ensured using the Lyapunov theory. The method has only been tested by simulations.

To avoid the complexity of accurately modeling the wire deformation behavior, model-free methods have been proposed. For instance, a method to automatically control the shape of a deformable object based on the online estimation of pseudo-stiffness matrices has been proposed in [Nav+16]. However, the method does not handle well wire manipulation when the wire presents a slight bending, or cases where the feature points are not well distributed on the object.

A method based on a novel online Gaussian Process Regression has been proposed in [HSP18]. The method requires a preliminary learning phase during which the velocities of a set of feature points are acquired with the velocities of the corresponding contact points. This training set can be heavy depending on the manipulation.

A learning-based approach relying on the Coherent Point Drift (CPD) method has been used both for wire tracking, task planning and trajectory planning in [TWT18]. The CPD method permits to compute the non-rigid transformation of a point set in another set. By

comparing the current tracked state of the wire with predetermined wire states stored in a database, it is possible to determine the next manipulation command to apply.

Other planning methods have been proposed, such as the method of Zhu *et al.* to perform wire-routing tasks [Zhu+19]. The planner takes both visual and force measurements as inputs. The planner controls two effectors in position: the first is mobile and moves one extremity of the wire, the other can only release or hold more firmly the second extremity. This approach relies on a preselected set of contact points with the environment to achieve a desired configuration.

Wire shape control has also been addressed using visual servoing methods. For instance, an adaptive model-free control method for catheter-like soft robot has been proposed in [Ver+19]. This method permits to align the eye-in-hand camera with an optical target when the catheter is inserted into a constrained environment. This method relies on several assumptions: viscoelastic and frictional losses are neglected and shearing is ignored.

Another method for controlling a 3-**DOF** catheter-like soft robot has been proposed in [Qi+14]. This method allows the tip of the catheter to converge to a desired position. To do this, six arbitrary operating points are selected anywhere on the catheter. For each of them, a local controller computes the control law to achieve this objective. A fuzzy controller mixes the results of the six controllers to generate the global control law. The system was only tested on Matlab simulation.

More recently, a method to control the 2D shape of a wire using several effectors has been proposed by Zhu *et al.* [Zhu+18]. Each effector has 3 **DOFs** (2 translations, 1 rotation). The wire shape is approximated by a Fourier series. The geometric deformation model is adapted online, assuming that a change in the Fourier coefficients depends linearly on a change in robot position. Sets of changes in Fourier coefficients and sets of changes in robot position are continuously acquired. These sets are used in a least-squares minimization to compute the Jacobian matrix that relates the robot motions to the changes of shape. However, if the robot did not move in all the affordable directions, the Jacobian matrix estimation update is delayed to avoid any singularity. The Jacobian matrix is finally used to compute the control law. The method was tested using a wire held by two grippers.

4.2.3 Our contributions

Our contributions to the automatic shape control of deformable wires are as follows: the definition of a new kind of visual feature permitting to model the 3D shape of a wire , a marker-less model-free tracking method that permits to track at an interactive time a deformable wire and a novel shape servoing method to control its deformations.

The visual feature that we propose does not require any mechanical properties of the wire to infer its 3D shape. Instead, it relies on a geometric model based on B-splines [De 72]. The use of B-splines allows our method to control the 3D shape of a wire, while the method proposed in [Zhu+18] only handled 2D shapes since this latter uses 2D Fourier series. Moreover, it permits to handle small bending situations and to ensure that the feature points are evenly distributed over the length of the wire when it is in its resting state, contrarily to the method proposed in [Nav+16].

Unlike the tracking methods mentioned above, such as [TT18] or [Sch+13], the tracking method that we propose does not require any marker or mechanical model to track the 3D shape of a wire that undergoes deformations. Instead, it relies on Sequential Importance Resampling (SIR) particle filtering [GSS93]. The use of SIR particle filtering permits to determine which B-spline best matches visual observations provided by a **RGB-D** camera.

The shape control is carried out by a closed-loop visual servoing. Two end-effectors holding the wire extremities are controlled to make the wire converge to a desired 3D shape. The control law is computed using a model-free shape servoing method derived from the ADVISEd method presented in Chapter 3. Unlike [HSP18], the shape servoing method that we propose does not require a complex initialization phase. The Jacobian matrix that relates the motions of the end-effectors to the deformation of the wire is initialized by small motions in every direction the system can afford and is adapted online using weighted least-squares minimization with sliding window, as presented in Chapter 3. A confidence threshold permits to update solely the parts of the Jacobian matrix impacted by the most recent directions of motion of the robots, while the method proposed in [Zhu+18] does not recompute the Jacobian matrix when the robots did not moved in all the affordable directions.

4.3 Methodology of the automatic shape control of deformable wires

The challenge addressed in this chapter is the automatic 3D shape control of deformable wires. To address this challenge, we first propose a novel kind of visual feature that permits to infer the 3D shape of a homogeneous deformable wire based on a geometric model . Secondly, we propose a marker-less model-free deformable wire tracking method based on particle filtering. Finally, we propose a visual servoing approach to control two end-effectors holding the extremities of the wire. This approach has the advantage of not relying on any a priori mechanical model.

In this section, we first explain the novel kind of visual feature to represent the current 3D

shape of a deformable wire from a point cloud. Then, we present the deformable wire tracking method. Finally, we present the active shape servoing method.

4.3.1 Visual feature based on a geometric model

We propose a novel visual feature that permits to infer the 3D shape of a homogeneous deformable wire from a point cloud using a **RGB-D** camera. The shape of a wire is represented by a B-spline geometric model. A B-spline is a piecewise polynomial function with user-defined order θ. A B-spline can be constructed from a set of control points $\mathbf{c}_j$, whose number n is defined such as $n \leq \theta + 1$. These control points divide the B-spline into segments.

Let $\psi(h)$ be a B-spline, $\tilde{\mathbf{p}}(h) \in \mathbb{R}^3$ a point of a B-spline and $h \in [0;1]$ the normalized curvilinear coordinate of the B-spline. The relationship that exists between the B-spline point and the B-spline is as follows:

$$\tilde{\mathbf{p}}(h) := \psi(h) = \sum_{j=0}^{n-1} \mathbf{c}_j B_{j,\,\theta}(h), \text{ with } h \in [0;1] \tag{4.1}$$

where $B_{j,\,\theta}(h)$ are the B-splines basis functions that are defined recursively as follows:

$$B_{j,\,0}(h) := \begin{cases} 1 & \text{if } t_j \leq h < t_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

$$B_{j,\,k} := \frac{h - t_j}{t_{j+k} - t_j} B_{j,\,k-1}(h) + \frac{t_{j+k+1} - h}{t_{j+k+1} - t_{j+1}} B_{j+1,\,k-1}(h) \tag{4.3}$$

with t_j being the knot values defining the B-spline, i.e. the values of h that correspond to the control points $\mathbf{c}$. The knot values are usually gathered into a knot vector $\mathbf{t}$ and are defined such as:

$$\begin{cases} & 0 = t_0 \leq t_1 \leq t_2 \cdots \leq t_{n-1} \leq t_n = 1 \\ \mathbf{t} = & \Big(t_0 \ldots t_0 \; t_1 \; t_2 \ldots t_n \ldots t_n\Big) \text{ with } n-1 \text{ repetitions of } t_0 \text{ and } t_n \end{cases} \tag{4.4}$$

For the sake of brevity, a B-spline point will be thereafter designated by $\tilde{\mathbf{p}}$ and a B-spline by ψ.

We propose to use η equidistant B-spline points to construct the deformation feature vector $\mathbf{s}$ used in the control law. Assuming that the deformable wire is homogeneous, the use of equidistant B-spline points as set of deformation features ensures a registration between the physical points of the wire and the set of deformation features over time. Indeed, when the

length of the physical wire changes due to stretching, the same physical points become more distant from each other than they were in the resting state. The same behavior is observed with the equidistant B-spline points, because the normalized parametric length now represents a greater length of wire. To generate η equidistant B-spline points, we proceed as follows:

$$\tilde{\mathbf{p}}_i = \psi(\frac{i}{\eta - 1}), \text{ with } i \in \mathbb{N},\ i \in \{0;\ \eta - 1\} \tag{4.5}$$

4.3.2 Deformable wire tracking method

We propose to use a SIR particle filter to track the wire. A particle filter generates χ particles, also known as sample states, representing the internal state of an observed system. At each time step, it updates each sample state according to a predictive update rule. Once updated, the likelihood between each sample state and the observations of the system is computed. Finally, a weighted sum of all the sample states is performed to generate an estimate of the internal state of the observed system. The weight of a sample is proportional to its likelihood.

In our case, the SIR filter sample states are B-splines and the observation is a segmented 3D point cloud of the wire.

Update of a particle

Let $\mathbf{p}$ be an observed point of the wire in the point cloud. Let $\mathbf{m}_{a\mathbf{c}_j}$ and $\mathbf{m}_{b\mathbf{c}_j}$ be the influence of the motion of each end-effector on the predictive motions of a control point $\mathbf{c}_j$ of a B-spline. The influence of the end-effectors motions on the linear velocity of the control point $\mathbf{c}_j$ is approximated using the screw theory. Then, by multiplying the linear velocity of the control point $\mathbf{c}_j$ by the sampling time, we get its corresponding 3D displacement. The influence of the motion of the first end-effector on the linear velocity of the control point $\mathbf{c}_j$, denoted by $\mathbf{m}_{a\mathbf{c}_j}$, can be written as follows:

$$\mathbf{m}_{a\mathbf{c}_j} = (\boldsymbol{v}_a + (\mathbf{c}_j - \tilde{\mathbf{p}}_0) \wedge \boldsymbol{\omega}_a)\Delta t \tag{4.6}$$

and the influence of the motion of the second end-effector on the linear velocity of the control point $\mathbf{c}_j$, denoted by $\mathbf{m}_{b\mathbf{c}_j}$, can be written as follows:

$$\mathbf{m}_{b\mathbf{c}_j} = (\boldsymbol{v}_b + (\mathbf{c}_j - \tilde{\mathbf{p}}_{\eta-1}\ ) \wedge \boldsymbol{\omega}_b)\Delta t \tag{4.7}$$

where:

- $\boldsymbol{v} = \begin{pmatrix} v_x, v_y, v_z \end{pmatrix}^T$ denotes the linear velocity of an end-effector while $\boldsymbol{\omega} = \begin{pmatrix} \omega_x, \omega_y, \omega_z \end{pmatrix}^T$

denotes its angular velocity. The subscript "a" represents the first end-effector and the "b" the other one,

- $\mathbf{a} \wedge \mathbf{b}$ is the cross-product between two vectors,
- Δt is the robot control law sampling time.

The control point $\mathbf{c}_0$ that corresponds to the extremity of the wire attached to the end-effector "a" is updated using directly $\mathbf{m}_{a\mathbf{c}_j}$. The same update rule is used for the control point $\mathbf{c}_{n-1}$ that corresponds to the extremity of the wire attached to the end-effector "b", using $\mathbf{m}_{b\mathbf{c}_j}$ instead of $\mathbf{m}_{a\mathbf{c}_j}$.

For the other control points, the update rule that we propose to compute the new position of each control point $\mathbf{c}_j$ of a B-spline model is a linear combination of the influences of the two end-effectors on the control point motion, namely $\mathbf{m}_{a\mathbf{c}_j}$ and $\mathbf{m}_{b\mathbf{c}_j}$. The closer a control point is to an end-effector the greater the weight assigned to this end-effector influence is. Additionally, to give more weight to the end-effector motions that are in the direction of the wire orientation, we propose to compute the scalar product between the tangent to the control point $\mathbf{t}_{\mathbf{c}_j}$ and the end-effectors influences $\mathbf{m}_{a\mathbf{c}_j}$ and $\mathbf{m}_{b\mathbf{c}_j}$. The update rule that we propose to compute the new position of each control point $\mathbf{c}_j$ of a B-spline model is therefore the following:

$$\mathbf{c}_j(k) = \mathbf{c}_j(k-1) + (\mathbf{t}_{\mathbf{c}_j} \cdot \mathbf{m}_{a\mathbf{c}_j})\frac{\theta - j}{\theta}\mathbf{m}_{a\mathbf{c}_j} + (\mathbf{t}_{\mathbf{c}_j} \cdot \mathbf{m}_{b\mathbf{c}_j})\frac{j}{\theta}\mathbf{m}_{b\mathbf{c}_j} + \boldsymbol{\nu},\ j \in \{1; n-2\} \tag{4.8}$$

where:

- k is the current discrete time,
- $\mathbf{t}_{\mathbf{c}_j}$ is the tangent of the B-spline at the control point $\mathbf{c}_j$,
- $\boldsymbol{\nu}$ is a vector of Gaussian noise with a zero mean and a standard deviation $\sigma_{\boldsymbol{\nu}}$,
- θ is the order of the B-spline,
- $\mathbf{a} \cdot \mathbf{b}$ is the scalar product between two vectors.

The whole B-spline model is updated accordingly to the new position of its control points. The update of a B-spline model is illustrated in Figure 4.1.

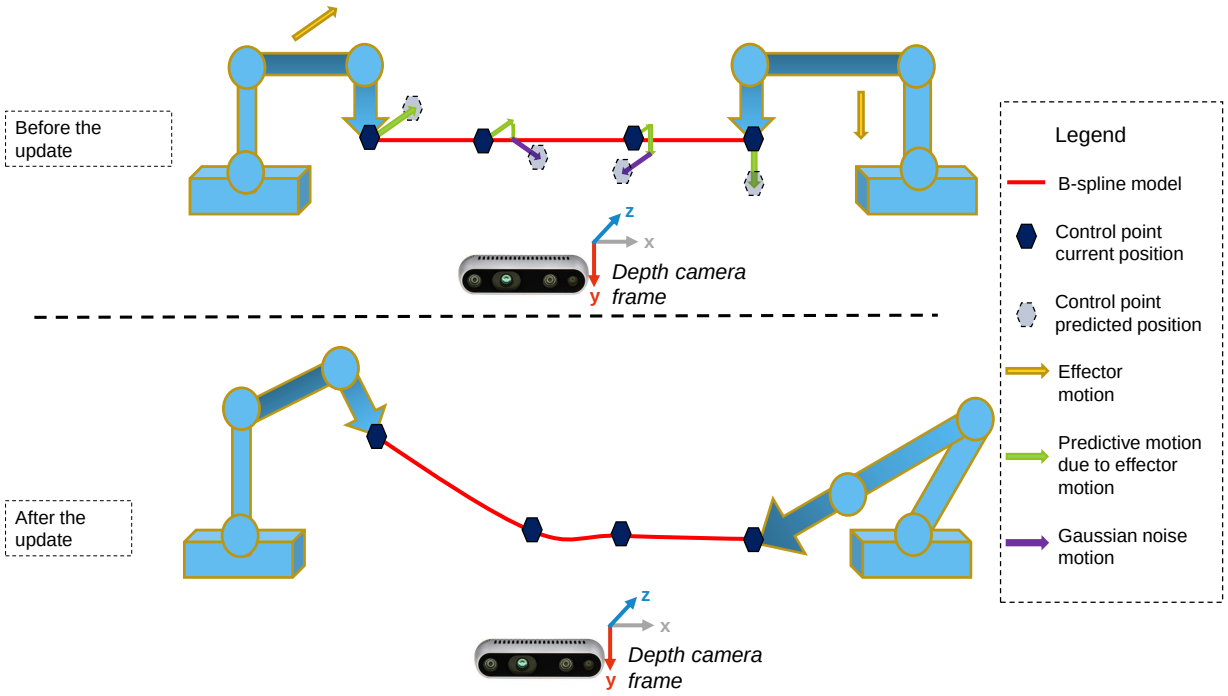


Figure 4.1: 2D illustration of the update of a particle. (Top) The wire B-spline continuous model is depicted by a red line, B-spline control points by dark blue hexagons and motions by arrows in different colors. The end-effector motions are depicted by yellow arrows. The predictive motion of the control points due to end-effectors motions is depicted in green arrow whose length is equal to their weight. The additive Gaussian noise is represented by purple arrows. The pale dotted hexagons represent the future position of each control point once the update will be complete. (Bottom) Illustration of the particle state after the update.

Likelihood evaluation

Let $\mathbf{p}_{closest}$ be an observed point in the point cloud that is the closest from a B-spline point $\tilde{\mathbf{p}}$ The likelihood of a B-spline point $w(\tilde{\mathbf{p}})$ is evaluated on a window of the depth map taking into account $\Omega \times \Omega$ neighboring points, as depicted in Figure 4.2. The further a neighboring point is from the evaluated point $\tilde{\mathbf{p}}$ the smaller is its weight in the likelihood computation. The window selects neighboring points in the plan normal to the wire and orthogonal to the camera optical axis because a depth camera generates a depth map that does not permit to see behind an object: considering a three dimensional window would lead to artificially reducing the likelihood of a B-spline point due to occlusion.

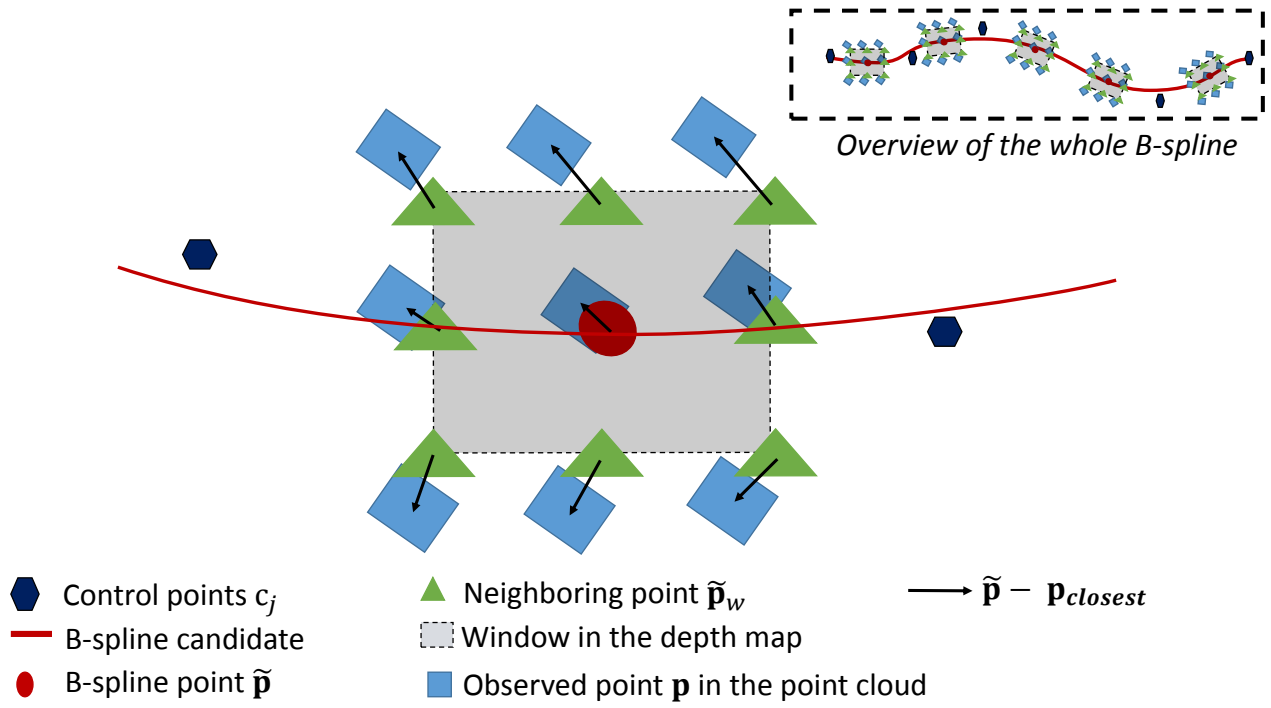


Figure 4.2: 2D illustration of the likelihood computation. The wire B-spline continuous model is depicted by a red line, B-spline points by red circles and B-spline control points by black hexagons. The triangles represent the B-spline neighboring points. The squares represent the observed points in the point cloud. The arrows represent the vectors that link a B-spline point to its closest point in the point cloud. An overview of the entire wire is depicted in the top right corner of the illustration.

Let $\mathbf{G} \in \mathbb{R}^{\Omega,\,\Omega}$ be the Gaussian weight matrix of standard deviation σ_w defined by:

$$\mathbf{G}_{i,j} = \frac{1}{2\pi\sigma_w^2} \exp\left\{-\frac{(i-1-A)^2 + (j-1-A)^2}{A^2}\right\} \tag{4.9}$$

where $A \in \mathbb{N}$ such as $A \leq \frac{\Omega}{2} < A+1$.

Let g be a Gaussian function of standard deviation σ_g that represents the similarity between a B-spline point $\tilde{\mathbf{p}}_i$ and its closest observed point $\mathbf{p}_{closest}$. It is defined by:

$$g(\tilde{\mathbf{p}}, \mathbf{p}_{closest}) = \frac{1}{\sigma_g\sqrt{2\pi}} \exp\left\{-\frac{\|\tilde{\mathbf{p}} - \mathbf{p}_{closest}\|^2}{2\sigma_g^2}\right\} \tag{4.10}$$

The likelihood of a B-spline point $w(\tilde{\mathbf{p}})$ is defined by:

$$w(\tilde{\mathbf{p}}) = \frac{1}{\sum_{i,j=1}^{\Omega} \mathbf{G}_{i,j}} \sum_{i,j=1}^{\Omega} \mathbf{G}_{i,j} g(\tilde{\mathbf{p}}_w(i,j), \mathbf{p}_{closest}) \tag{4.11}$$
$$\text{with } \tilde{\mathbf{p}}_w(i,j) = \tilde{\mathbf{p}} + 2\sigma_w(\frac{i-1}{A} - 1)\mathbf{t}_{\tilde{\mathbf{p}}} + 2\sigma_w(\frac{j-1}{A} - 1)\mathbf{n}_{\tilde{\mathbf{p}}}$$

where $\mathbf{t}_{\tilde{\mathbf{p}}}$ is the tangent of the B-spline at the wire point and $\mathbf{n}_{\tilde{\mathbf{p}}}$ is the normal, located in the XZ plane of the depth camera, of the B-spline at the wire point.

The likelihood of a B-spline $W(\psi)$ is defined as:

$$W(\psi) = \frac{\sum_{i=1}^{\eta} w(\tilde{\mathbf{p}}_i)}{\eta} \tag{4.12}$$

We obtain the optimal B-spline that represents the current state of the wire resulting from the SIR filter $\hat{\psi}$ by:

$$\hat{\psi} = \frac{1}{\sum_{i=1}^{\chi} W(\psi_i)} \sum_{i=1}^{\chi} W(\psi_i)\psi_i \tag{4.13}$$

An arbitrary 3D point of the optimal B-spline is denoted $\hat{\tilde{\mathbf{p}}}_j$.

4.3.3 Adaptive shape servoing method

Let $\mathbf{s}$ be the current deformation feature vector that represents the 3D shape of a wire and $\mathbf{s}^*$ the desired shape. As explained in Section 4.3.1, we propose to use η equidistant B-spline points to construct the deformation feature vector $\mathbf{s}$ used in the control law. The deformation feature vector is defined as follows:

$$\mathbf{s} = \left(\hat{\tilde{\mathbf{p}}}_{1x}, \hat{\tilde{\mathbf{p}}}_{1y}, \hat{\tilde{\mathbf{p}}}_{1z}, \ldots, \hat{\tilde{\mathbf{p}}}_{\eta x}, \hat{\tilde{\mathbf{p}}}_{\eta y}, \hat{\tilde{\mathbf{p}}}_{\eta z}\right)^T \in \mathbb{R}^{3\eta} \tag{4.14}$$

Let define m as the sum of the number of DOFs of the end-effectors, which is 12 in our case since two 6-DOF robotic arms are used to manipulate the wire. Let $\mathbf{J}$ denote the deformation Jacobian that relates the variation of the 3D shape of the wire with the end-effectors velocity $\mathbf{v^e} = \left({\boldsymbol{v}_a}^T, {\boldsymbol{\omega}_a}^T, {\boldsymbol{v}_b}^T, {\boldsymbol{\omega}_b}^T\right)^T$ such that $\dot{\mathbf{s}} = \mathbf{J}\mathbf{v^e}$.

In our shape servoing method, we propose to online estimate the deformation Jacobian matrix using the adaptive method presented in Section 3.2.2.

The velocity control law finally applied to control the two end-effectors is defined by $\mathbf{v^{ctrl}} = -\alpha\hat{\mathbf{J}}^+\left(\mathbf{s} - \mathbf{s}^*\right) \in \mathbb{R}^m$ where $\alpha \in \mathbb{R}^+$ denotes the control law gain and $\hat{\mathbf{J}}^+$ denotes the

Moore-Penrose pseudo-inverse of the estimated deformation Jacobian $\hat{\mathbf{J}}$.

4.4 Experiments

Different experiments have been conducted. First, experiments have been conducted in order to evaluate the impact of the robots motions responsible for the deformation of the wire on the performances of the method. Then, experiments have been conducted in order to evaluate the impact of the wire material on the method performance. Another series of experiments have been conducted to evaluate the impact of the geometric model resolution on the throughput of the method. Finally, various experiments were conducted in order to use the method in order to generate large deformations.

4.4.1 Experimental Setup

The experimental setup, shown in Figure 4.3, is composed of two 6-**DOF** anthropomorphic robot arms (Viper 650 and 850 from ADEPT). Their end-effectors permit to attach a deformable wire. The length of the deformable wires section that is considered during the experiments is around 150mm.

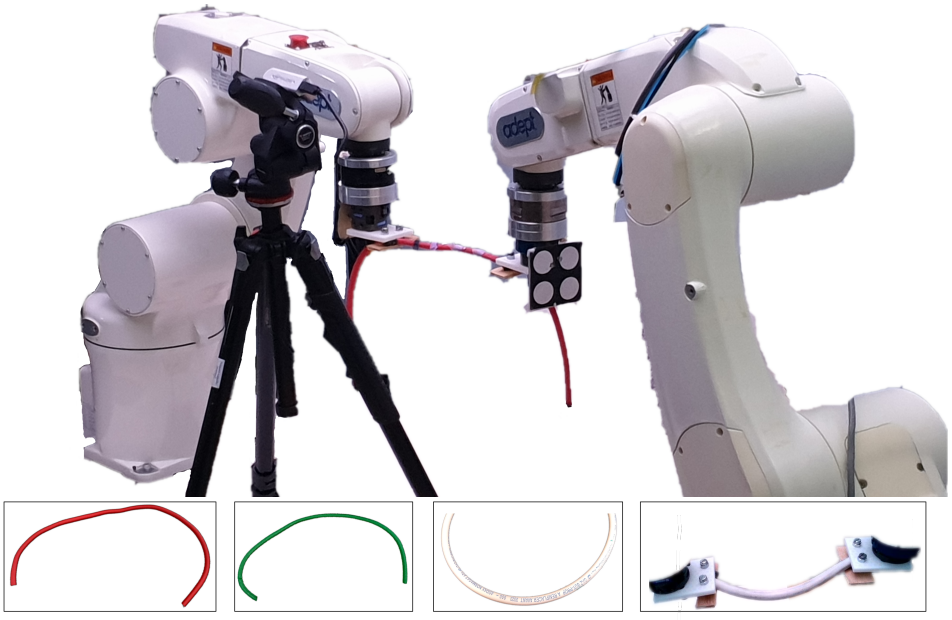


Figure 4.3: (top) Experimental setup composed of two robot arms that manipulate a wire tracked by a **RGB-D** camera. (bottom) The different types of wires that were tested and the fixation system.

An Intel Realsense D435 **RGB-D** remote camera is used to track the deformable wire. For all the experiments, the wire was placed about 400mm from the **RGB-D** camera, the depth information is therefore accurate within to ±0.6mm [Int20]. To improve the accuracy of depth measurements, the texture of the wire has been augmented by randomly sticking colored pieces of paper. These are not used for tracking. The color and depth segmentation is performed using the PCL library [RC11] to isolate the wire from the rest of the observed point-cloud.

Our method has been implemented on a computer equipped with an Intel Xeon CPU running at 3.70GHz with 16 logical cores.

To define a desired wire shape, the robots are moved before an experiment and the resulting shape is recorded. The robots and the wire are then returned to their resting state, common to each experiment and depicted in Figure 4.4.

Figure 4.4: Resting state common to every experiment.

To initialize the deformation Jacobian, a preliminary open-loop procedure is performed where both end-effectors move successively in each affordable direction over a very short distance while measuring the resulting deformation variation of the wire. The initial values of the components of the deformation Jacobian are then computed numerically by finite difference. This preliminary excitation being carried out for each **DOF** of the system, the initial value of the deformation Jacobian is therefore full rank. Moreover, it will maintain full rank during the shape visual control thanks to the Jacobian update Equation (3.12) that is performed only for sufficiently excited directions.

For all the experiments, the parameters were set empirically as follows: α= between 0.0125 and 0.025, Υ= 0.99, N= 10 and ϵ = 0.001. For the SIR particle filter parameters: η= 50, $\chi = 50$, $\theta = 3$, number of segments = 3. The control law sample time Δt is set to 40ms.

4.4.2 Performances evaluation criteria

The method is evaluated with regard to several criteria:

- the average displacement between the undeformed wire shape and the one at the final discrete time k_f defined by:

$$d_{av} = \frac{\sum_{i=1}^{\eta} \|\tilde{\mathbf{p}}_i(k_f) - \tilde{\mathbf{p}}_i(0)\|}{\eta} \tag{4.15}$$

- the average error e_{Av}, which is the average of the Euclidean distance between a current wire shape point and its desired position. It is defined by:

$$e_{Av}(k) = \frac{\sum_{i=1}^{\eta} \|\tilde{\mathbf{p}}_i(k) - \tilde{\mathbf{p}}_i^*\|}{\eta} \tag{4.16}$$

- the relative norm error expresses the ratio of the difference between the expected deformation and the final one. It is expressed in percentage and defined by:

$$\overline{e_n} = \frac{100}{\eta} \sum_{i=1}^{\eta} \frac{\|\tilde{\mathbf{p}}_i^* - \hat{\tilde{\mathbf{p}}}_i(0)\| - \|\hat{\tilde{\mathbf{p}}}_i(k_f) - \hat{\tilde{\mathbf{p}}}_i(0)\|}{\|\tilde{\mathbf{p}}_i^* - \hat{\tilde{\mathbf{p}}}_i(0)\|} \tag{4.17}$$

- the median duration is the duration required to acquire the data from the camera, perform the tracking of the wire, update of the Jacobian and compute the control law. It is defined by:

$$t_{med} = median_{k=0}^{k_f}\Big(\Delta t(k)\Big) \tag{4.18}$$

The convergence duration presented in Tables 4.1 and 4.2 corresponds to the duration that was required to make the shape of the wire converge towards the desired one.

4.4.3 Influence of the type of robots motions

We first evaluate the performances of the method with respect to the robots motions responsible for the deformation of the wire. Shape servoing involving translation only, rotation only and a combination of rotations and translations of the two end-effectors have been tested. Table 4.1 presents the results of these experiments. The method was able to converge below 5mm in terms of average error e_{Av} and below 13% in terms of relative norm error $\overline{e_n}$ for each translation-based experiment. It was more difficult to notice deformations using only the X-axis and the Z-axis

camera frame rotation-based motions of the end-effectors as the resulting deformation occurred mainly in the Z-axis of the camera frame. The measurements in this direction are approximate due to the low quality of the **RGB-D** sensor in the Z-direction (depth). Consequently, the relative norm error is greater for the rotation-based experiments. For the combined rotations-translations experiments, the method was able to converge to 2.8mm in terms of average error and below 12% in terms of relative norm error. In almost all experiments, the average displacement is lower than the average initial error. This indicates that the method did not over-deform the wire during the experiments. For all the experiments, the median duration of computation process was about 20ms, proving that the method can work at an interactive time.

Table 4.1: Results of experiments where the deformation resulted from different robots **DOF** motions. The four first columns respectively give the average displacement, the relative norm error, the initial average error and the final average error. The fifth column indicates the duration required to make the wire converge to the desired shape. The last column indicates the robots motions that generated the desired deformed shape. “tx", “ty" and “tz" correspond to translations along each axis, while “rx", “ry" and “rz" correspond to rotations. The last three lines correspond to a combination of translations and rotations.

d_{av} **(mm)**	$\overline{e_n}$ **(%)**	$e_{Av}(0)$ **(mm)**	$e_{Av}(k_f)$ **(mm)**	**Convergence duration (s)**	**Robots motions**
17.6	9.3	17.5	2.5	70	tx
10.5	10.6	10.3	2.7	170	ty
11.7	12.2	12.2	3.6	70	tz
12.9	27.3	11.2	4.7	190	rx
11.6	15.7	11.6	2.1	70	ry
9	20.6	8	4.1	50	rz
13.2	12.9	15.1	4.7	150	txtytz
16.2	8.6	15.5	2.8	130	rztxtytz
12.6	11.4	12.7	2.8	70	rxryrztxtytz

4.4.4 Influence of the wire nature

We tested the independence of the method with respect to mechanical and visual properties of the wire. We performed several experiments using wires of different materials, diameters and colors: two twined elastic ropes of eight-millimeter diameter in different colors and a 13-

millimeter diameter flexible pale grey gas pipe. The different objects are presented in Figure 4.3 as insets. The experiments combined deformations due to the rotation along the Z axis and translations along each axis of the two end-effectors Cartesian frame. Figures 4.5a to 4.5c illustrate the final shape of the deformable wire projected on the XY, XZ, YZ planes in order to observe the 3D deformation for each of these experiments, as well as the evolution of the average error over time. In these figures, the relative norm error is coded according to a color scale: the greener the wire the closer to 0% the error is, the redder the wire the closer to 30% the $\overline{e_n}$ error is. From similar initial and desired shapes, the method was able to converge with a similar accuracy: the final average error is 3.3mm for the red wire, 4mm for the green wire and 2.9mm for the gas pipe. However, one can notice in the different plan projections that the gas pipe converged to a smoother shape than the elastic ropes. The relative norm error is also lower in the gas pipe experiment: 3.9% against 10.6% for the red wire and 9.4% for the green wire. The material of which it is made of may have induced smoother deformations that are therefore easier to control than the twined ropes. The median duration of computation process was about 17ms for each wire.

(a) First rope

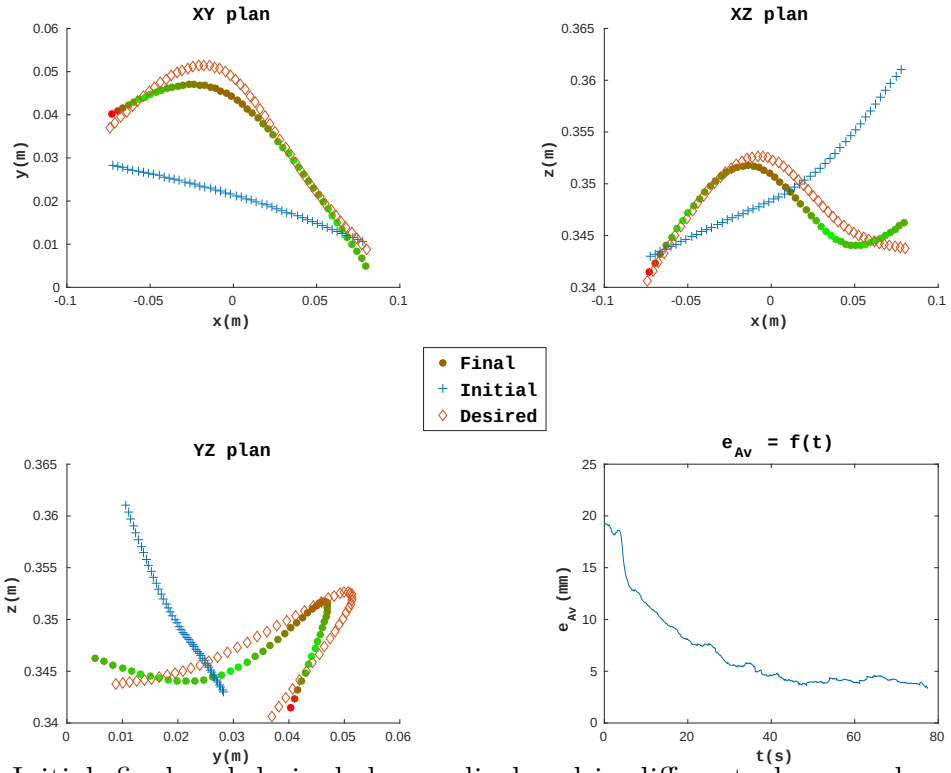
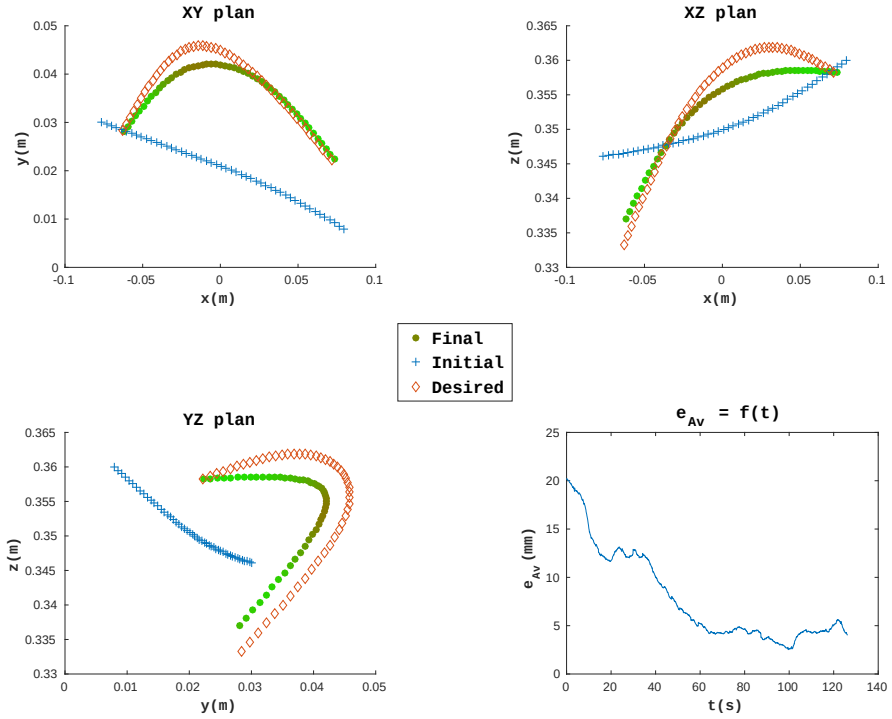


Figure 4.5: Initial, final and desired shapes displayed in different planes and evolution of the average error over time for the manipulations of three different wires (a) Red rope (b) Green rope (c) Gas pipe, made of different materials and colors.

(b) Second rope



(c) Gas pipe

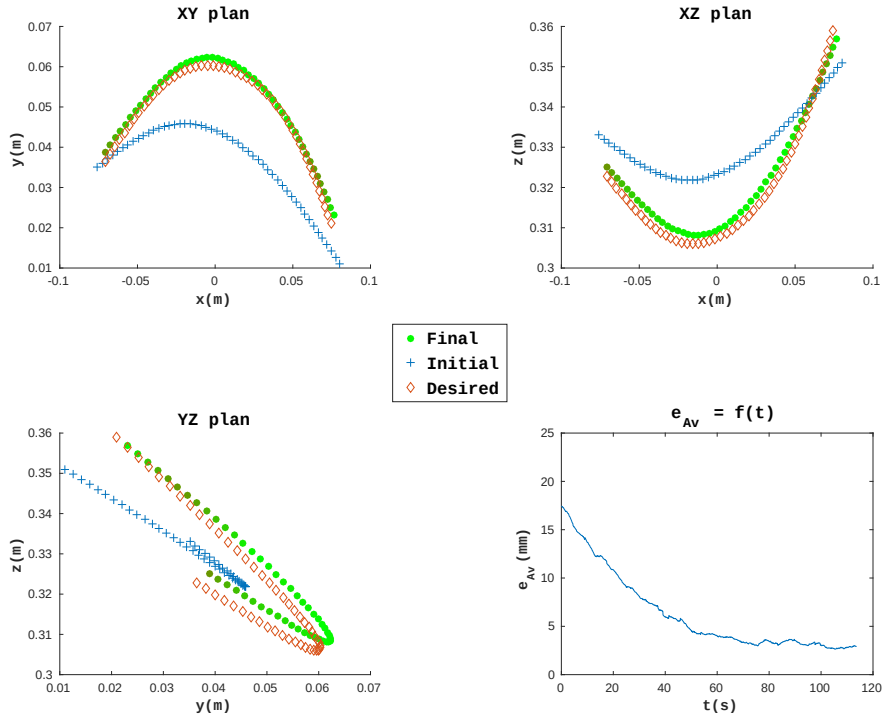


Figure 4.5: (continued from previous page) Initial, final and desired shapes displayed in different planes and evolution of the average error over time for the manipulations of three different wires (a) Red rope (b) Green rope (c) Gas pipe, made of different materials and colors.

4.4.5 Influence of the resolution of the B-Spline

We also performed tests to determine if the accuracy of the method is related to the number of points η of the B-spline used to model the shape of the wire. Experiments were conducted combining translations along each axis of the two end-effectors Cartesian frame. The same initial and desired shapes were used for each of these experiments, which were performed using $\eta = 10,\ 25,\ 50,\ 75$ and 100 points respectively. Table 4.2 presents the results of these experiments. Starting from similar initial and desired shapes, the method was able to converge with similar accuracy: the final average error is 3.5mm for 10 points, 3.2mm for 25 points, 4.6mm for 50 points, 5.5mm for 75 points and 3.1mm for 100 points. In terms of time performance, the method still worked at an interactive time using $\eta = 100$, with a median duration of computation process of 19.8ms.

Table 4.2: Evaluation criteria with respect to the resolution of the B-Spline. The four first columns are the same as in Table 4.1. The fifth column indicates the median duration of computation process. The sixth column indicates the convergence duration defined as in Table 4.1. The seventh column indicates the number of points used to model the wire. For each experiment, a combination of translations along each axis of the robots has been performed, as shown in the last column.

d_{av} **(mm)**	$\overline{e_n}$ **(%)**	$e_{Av}(0)$ **(mm)**	$e_{Av}(k_f)$ **(mm)**	t_{med} **(ms)**	**Convergence duration (s)**	η	**Robots motions**
13.4	16.0	14.6	3.5	3.1	100	10	txtytz
15.3	13.2	15.2	3.2	5.6	140	25	txtytz
13.5	5.5	13.7	4.6	10.3	100	50	txtytz
13.3	16.2	15.8	5.5	14.8	90	75	txtytz
15.5	18.9	16	3.1	19.8	100	100	txtytz

4.4.6 Large deformation experiments

The deformation Jacobian is only valid for a small range of wire shape deformations. We carried out an experiment to determine if this could prevent the method from being used to generate large deformations. The initial shape of the wire is depicted as an inset in Figure 4.3, while the desired shape is depicted in Figure 4.6a. Figure 4.7 shows the evolution of the average error between the current shape and the desired one, as well as the initial, final and desired shapes of the wire displayed in the XY, XZ and YZ planes. The final wire shape uses the same color-

coded representation as in Section 4.4.4. One can notice that the method failed to converge towards the desired final shape. This is due to the fact that it fell into a local minimum at time t=25s. This motivated us to conduct experiments aiming at generating large deformations by considering a succession of desired intermediary shapes.

(a) Desired shape

(b) Final shape

Figure 4.6: Experiment with large deformations of the wire. As one can notice, the method failed to converge to the desired final shape as it fell into a local minimum during the experiment. Subdividing the task into several subtasks could have avoided this problem.

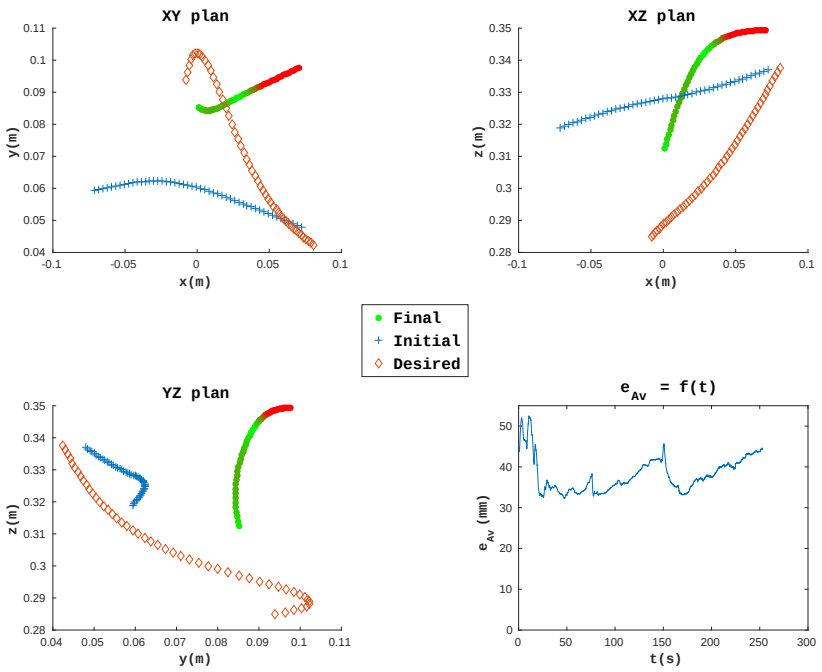


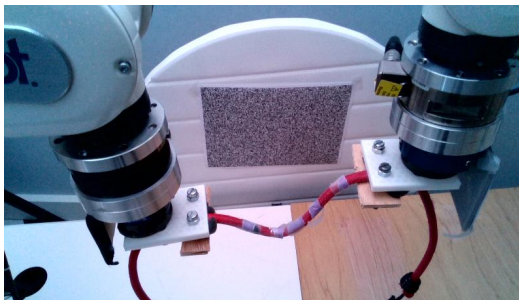
Figure 4.7: Large deformation case: initial, final and desired shapes shown in different planes and average error over time.

4.4.7 Large deformation experiments using intermediary targets

To enable our method to reach a desired deformation too different from the initial state, we can define successive intermediary targets. We carried out an experiment using this approach by defining before the experiment a trajectory composed of four desired intermediary shapes and a desired final shape. The intermediary shapes are chosen between the initial and desired shapes such as they describe a sequence from the furthest to the closest shape of the desired final shape, as depicted in Figure 4.8.

(a) 1^{st} intermediary target

(b) 3^{rd} intermediary target



(c) Final target.

(d) Final configuration

Figure 4.8: Intermediary targets for reaching large deformations of the wire. Three first pictures: some of the intermediary targets. Last picture: final target and final configuration reached at convergence.

The overall experiment lasted 380s. The median duration of the computation process was 16.6ms. The average displacement between the initial and final shape was 39.3mm. The initial average error was $e_{Av}(0) = 39$mm while the final one was $e_{Av}(k_f) = 3.9$mm. Figure 4.9 shows the evolution of the average error between the current shape and the desired final shape, as well as the initial, final and desired shapes of the wire displayed in the XY, XZ and YZ planes. The final wire shape uses the same color-coded representation as in Section 4.4.4. It is possible to notice that the average error between the current shape and the desired final shape steadily

decreased during the experiment. The change of intermediary targets did not disturb this convergence. It is possible to notice that the relative norm error is greater on the extremity of the wire where the torsion was the greatest. However, the relative norm error is only 6.1%, which proves that the method has indeed converged to the desired final shape.

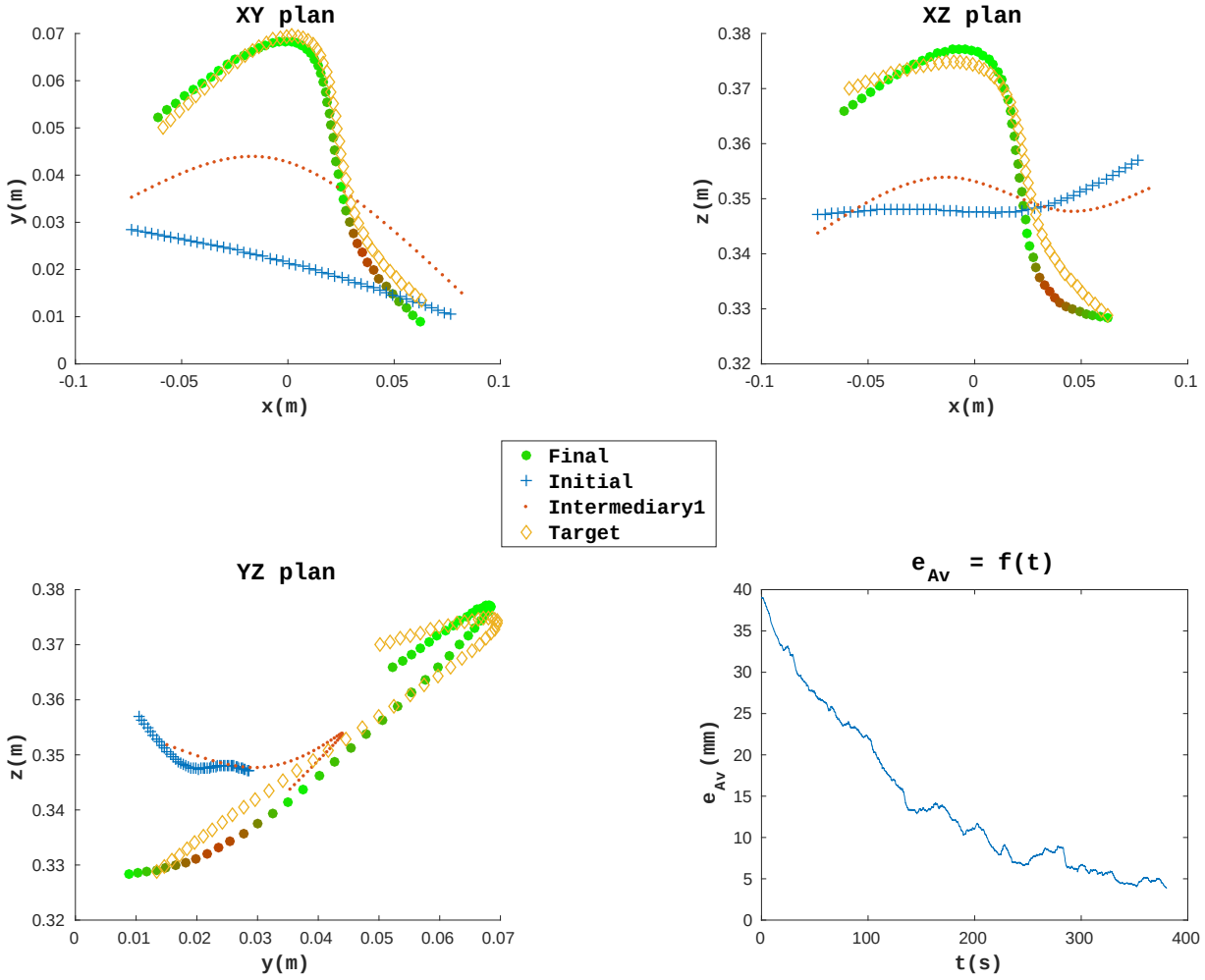


Figure 4.9: Large deformation with intermediary targets.

4.5 Conclusion

In this chapter, we addressed the challenge of the automatic 3D shape control of deformable wires. First, we proposed a novel kind of visual feature, relying on a B-spline geometric model, that permits to model the 3D shape of a deformable wire. Then, we proposed a marker-less model-free tracking method based on SIR particle filtering that permits to track deformable

wires. Finally, automatic shape control is ensured by an adaptive model-free shape servoing method. By controlling two end-effectors that hold both extremities of the wire, the method permits to make the wire reach a desired 3D shape. Different experiments have been conducted to evaluate the shape servoing method. A series of experiments has demonstrated that deformations resulting from translation-based and/or rotation-based motions of the end-effectors could be obtained with an average error below 5mm. A second series of experiments has demonstrated that the wire tracking and the control methods were independent of the mechanical properties of the wire. A third series of experiments has demonstrated that the method was able to work at an interactive time with different resolutions of the B-spline model of the wire. Finally, experiments have demonstrated that the method can be used to reach complex 3D shapes by defining intermediary targets.

Chapter 5

Simultaneous Tracking and Elasticity Parameter Estimation of Deformable Objects

Contents

5.1 Introduction

Nowadays, robots are efficient in manipulating rigid objects. However, simultaneous manipulation and tracking of deformable objects remain challenging and could improve different applications, such as humanoid robots interacting with soft objects, compliance testing in product manufacturing or detection of abnormal skin stiffness in preventive healthcare. In this chapter, we present an approach that automatically estimates the elasticity parameters of a soft object that is being deformed by the end-effector of a robot.

Our contributions are as follows:

- a model-based deformation tracking method that does not require any fiducial marker;
- an iterative elasticity estimation algorithm;
- a remote force estimation method.

The deformation tracking method and the elasticity estimation algorithm work in a closed-loop manner that allows to achieve two objectives: *STEPE* (Simultaneous Tracking and Estimation of Parameters of Elasticity) and *remote force estimation*. The required inputs of *STEPE* are a coarse 3D geometric model (mesh) of the object to track and the external measurements of the forces deforming the object. This work has been done jointly with A. Sengupta, PhD student, and has been published in [Sen+20].

The remainder of this chapter is divided into three sections. The next section details the methodology for the proposed method. The experimental results are presented next, followed by discussions and perspectives about our method. A video of the experimental evaluation is available at this link: https://youtu.be/3Ht-xwdaQrM.

5.2 Methodology of the STEPE and remote force estimation

5.2.1 Overview of the proposed method

Our goal is to simultaneously track a deformable object and estimate its elasticity parameters. To achieve this goal, we propose a closed-loop method depicted in Figure 5.1.

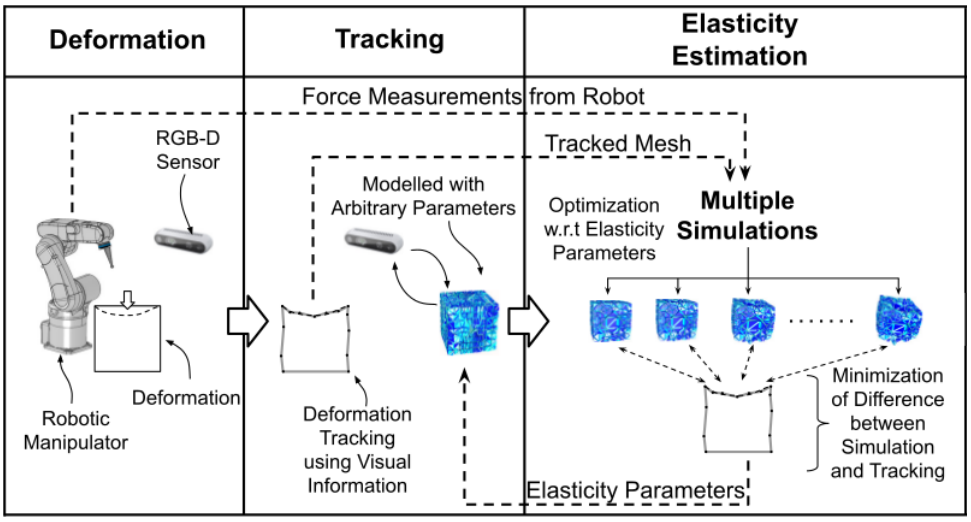


Figure 5.1: Overview of the STEPE architecture.

The deformation tracking method, represented by the **Deformation Tracking** block in Figure 5.1, uses an **RGB-D** camera to capture the deformation and tracks it using a physics-based simulation of the object deformations. The elasticity estimation algorithm, represented by the **Elasticity Estimation** block in Figure 5.1, uses the result of the deformation tracking method and measurements of the deformation forces, obtained from an external sensor, to estimate the elasticity parameters of the object. The external force sensor is mounted on the robot depicted in the **Deformation** block. Once the elasticity estimation is achieved, the parameters can be used by the deformation tracking method, thereby closing the loop. The method will iteratively converge towards a correct estimation of the elasticity parameters of the object. The elasticity parameters obtained at the convergence of the method are thereafter used for *remote force estimation*, represented by the **Force Estimation** block in Figure 5.2. From this stage, the deformation tracking method can be used to estimate the deformation forces acting on the object without any external force sensor, as illustrated in Figure 5.2.

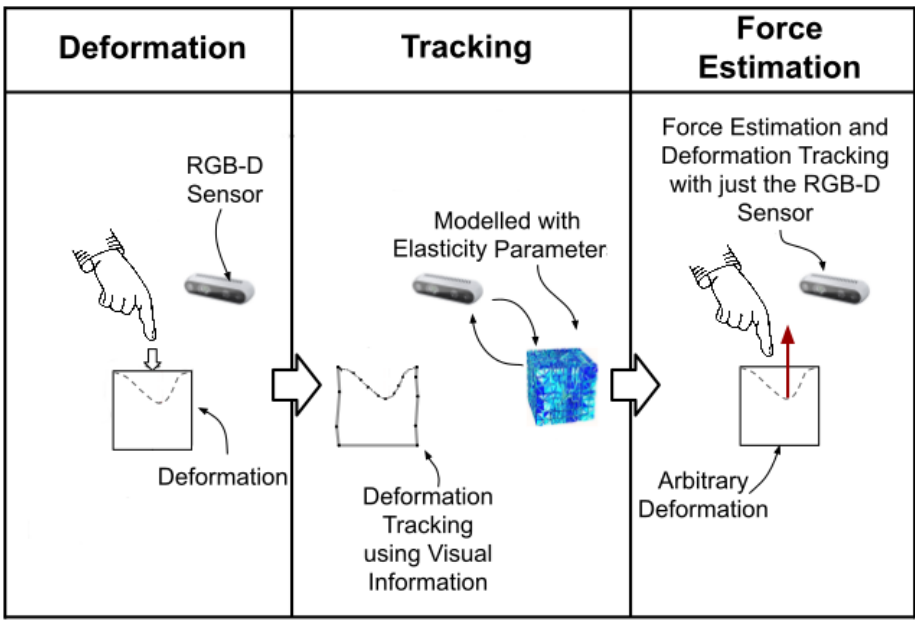


Figure 5.2: Overview of the system architecture when performing *remote force estimation*.

Modeling linear elastic objects

We chose to model the objects using the **Finite Element Method** (**FEM**) [Nea+06]. Objects are represented using tetrahedral meshes, such as the cube depicted in Figure 5.1. In order to avoid large rotational deformation artefacts, we use the **corotational FEM** variant of this method [MG04]. In this work, we use the *Hooke's law* that relates the stress and the strain applied to an object. This law uses a linear equation that depends on the elasticity parameters of the object, which are the Young modulus and the Poisson's ratio. This law is valid only for small deformations of linear elastic objects.

The linear algebraic equation of motion of a linear elastic object has been presented in Equation (2.22). This equation relates the deformations that the object undergoes to the external forces that it is subject to. The linear algebraic equation of motion of a linear elastic object is used to estimate both the elasticity parameters from deformation observations, and the external deformation forces from deformation observations when the elasticity parameters are available.

5.2.2 STEPE

Deformation of the object

To estimate the elasticity parameters of an object, it is necessary to have an accurate measurement of the deforming forces applied to the object as well as the visual observations of the resulting deformations. To ensure repeatability of experiments, we decided to use a robot equipped with a force sensor both to deform the object and to measure the resulting forces $\mathbf{f}^{meas} \in \mathbb{R}^3$. The robotic end-effector equipped with a force sensor and deforming the non-rigid object is depicted in the **Deformation** block of Figure 5.1, while Figure 5.3 shows the actual hardware used for the experiments. The end-effector deforms the object while recording the forces that are exerted. The end-effector motion is stopped at one point in order to maintain a static deformed state.

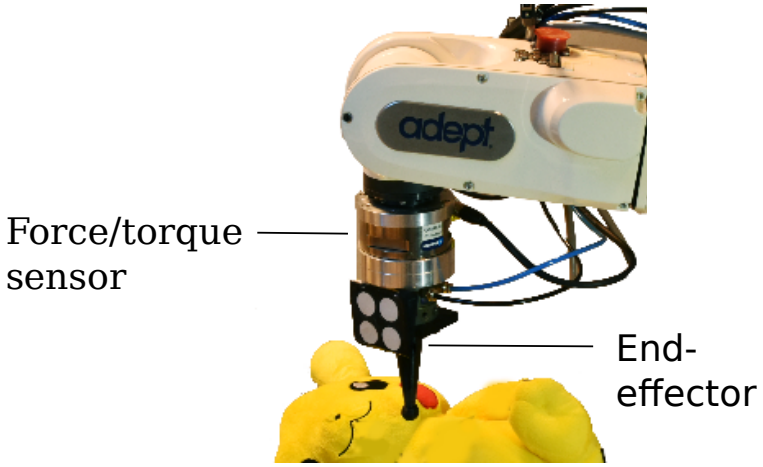


Figure 5.3: External force measurements hardware.

Deformation tracking method

We implemented a deformation tracking method similar to the one proposed in [SKM19]. A coarse 3D mesh of the object is used in a physics-based simulation. At the very beginning of the STEPE, the object is modeled using arbitrary elasticity parameters that are thereafter updated each time the elasticity parameter estimation converges, as depicted in the **Tracking** block of Figure 5.1. The principle of the deformation tracking is to determine external forces to apply on some vertices of the object mesh in the simulation in order to match the deformed mesh to the point cloud observed by a **RGB-D** camera. The vertices on which the forces must be applied are determined online using a clustering method and are called *control handles*

thereafter for the sake of brevity. Details on the clustering method can be found in [SKM19].

The initial pose of the object(O) with regard to the camera (C) is obtained in the first image frame using pre-trained markers [MUS16]. The pose of the object with regard to the camera, denoted by the homogeneous matrix ${}^{C}\mathbf{H}_{O}$, is updated at the beginning of each frame, using an approximate rigid object tracking method with the assumption that the object is rigid. This rigid tracking method consists in minimizing jointly a depth-based geometric error e^{D} and a keypoint-based error $\mathbf{e}^{K}$. Be ${}^{k}\mathbf{H}_{k-1}$ the initial estimate of the homogeneous matrix that transforms the object from its previous pose to its new pose. This homogeneous matrix encodes both the rotation matrix ${}^{k}\mathbf{R}_{k-1}$ and the translation vector ${}^{k}\mathbf{t}_{k-1}$. The jointly minimized cost functions for the approximate rigid tracking are given by:

$$e^{D}({}^{k}\mathbf{q}_{k-1}) = \left({}^{k}\mathbf{R}_{k-1}\,{}^{k-1}\mathbf{P} + {}^{k}\mathbf{t}_{k-1}\right) \cdot \mathbf{n}_{j} - d_{j} \tag{5.1}$$

$$\mathbf{e}^{K}({}^{k}\mathbf{q}_{k-1}) = \begin{pmatrix} x({}^{k}\mathbf{q}_{k-1}) - x^{*} \\ y({}^{k}\mathbf{q}_{k-1}) - y^{*} \end{pmatrix} \tag{5.2}$$

where ${}^{k}\mathbf{q}_{k-1} = ({}^{k-1}\mathbf{t}_{k}, \theta\mathbf{u})$ is the rigid motion that moved the object from its previous pose to the current one, such that θ and $\mathbf{u}$ are the angle and axis of the rotation ${}^{k}\mathbf{R}_{k-1}$ from the current image frame 'k' to the previous image frame. ${}^{k-1}\mathbf{P} = \left(X, Y, Z\right)$ represents an arbitrary 3D point from the last pointcloud, which has been matched to the j^{th} plane of the object's 3D model, denoted by the normal vector $\mathbf{n}_{j} = (n_{j}^{X}, n_{j}^{Y}, n_{j}^{Z})$ and distance to origin d_{j}. $(x({}^{k}\mathbf{q}_{k-1}), y({}^{k}\mathbf{q}_{k-1}))$ are the projection of the same arbitrary point in the $(k-1)^{th}$ image while (x^{*}, y^{*}) are the same image points matched in the k^{th} image using Harris corner features.

Once an approximate estimate of ${}^{k}\mathbf{q}_{k-1}$ has been obtained, the non-rigid tracking is done by applying external forces on the control handles in order to minimize the point-to-plane error, given by:

$$\mathbf{e}^{N}(\mathbf{P}) = \mathbf{n}_{j} \cdot {}^{k-1}\mathbf{P} - d_{j} \tag{5.3}$$

Let $\mathbf{J}$ be the Jacobian matrix that relates the variations of the point-to-plane error $\mathbf{e}^{N}$ to the variations of the external forces applied onto the *control handles* $\mathbf{F}$, i.e. $\dot{\mathbf{e}^{N}} = \mathbf{J}\dot{\mathbf{F}}$. This Jacobian matrix is computed using the method shown in Figure 5.4. Every *control handle* is subjected to a small force given by $\Delta\mathbf{F}_{x}$, $\Delta\mathbf{F}_{y}$ and $\Delta\mathbf{F}_{z}$, acting along the three axes. The displacements of all vertices of the mesh are computed by solving Equation (2.22), using an Euler-Implicit mechanism of a conjugate-gradient-based linear solver [Fau+12]. The Jacobian matrix $\mathbf{J}$ is then computed using finite differences in the value of $\mathbf{e}^{N}$ obtained from the three

deformed configurations. The force update is given by:

$$\Delta\mathbf{F} = -\boldsymbol{\lambda}\Big(\mathbf{WJ}\Big)^{+}\mathbf{We}^{N}({}^{k-1}\mathbf{P}) \tag{5.4}$$
$$\text{with } \boldsymbol{\lambda} = \begin{pmatrix}\lambda_X & \lambda_Y & \lambda_Z\end{pmatrix}$$

where $\mathbf{W}$ is a weighting matrix from Tukey based M-estimator [Mee+91], and $\boldsymbol{\lambda}$ is a scaling factor. The deformation caused in the mesh due to the Jacobian computation is discarded, and the update ${}^{k-1}\mathbf{F}+\Delta\mathbf{F}$ is applied to the *control handles*, where ${}^{k-1}\mathbf{F}$ is the external force already existing on the *control handles* at the previous image frame. This Iteratively Re-weighted Least Squares scheme is repeated for a fixed number of iteration at every image frame.

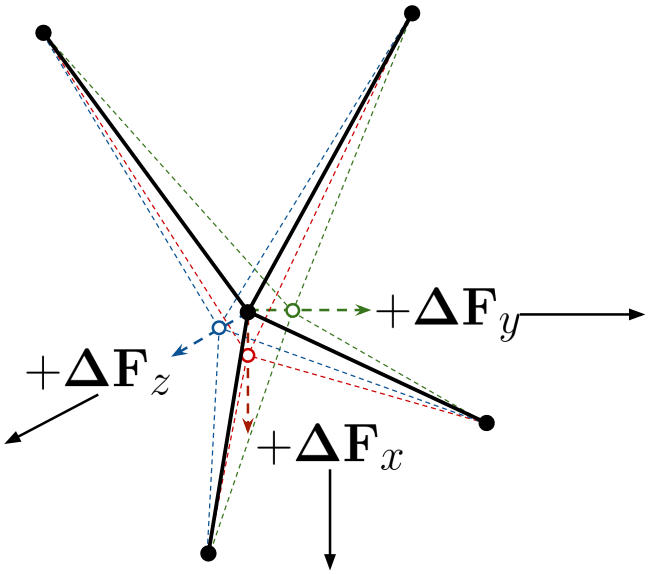


Figure 5.4: Every *control handle* is successively perturbed by a small force along one of the three axes. This permits to numerically compute the Jacobian matrix that permits to update the external forces applied onto the *control handles*, permitting the deformation tracking.

Elasticity parameter estimation algorithm

The **Elasticity Estimation** block of Figure 5.1 illustrates the principle of the elasticity parameter estimation algorithm. This estimation algorithm relies on an iterative scheme that minimizes the difference between the tracking results and physics-based simulations. The physics-based simulations use the Hooke's law evoked in Section 5.2.1 to represent the deformable objects. Consequently the elasticity parameters that can be estimated are the Young modulus and the Poisson's ratio. Let $\boldsymbol{\xi}$ be the elasticity parameter vector, V be the list of vertices, denoted

by v, that the camera can see, $\mathbf{f}^{meas}$ the deformation forces external measurements, $\mathbf{x}_v^t$ and $\mathbf{x}_v^s$ respectively the tracked and simulated 3D positions of v. The cost-function S that is minimized to estimate the elasticity parameters of the object at the contact point is given by:

$$S = \sum_{v \in V} \mathbf{e}_v^{\mathrm{E}}(\boldsymbol{\xi})^2 \tag{5.5}$$

In Equation (5.5), $\mathbf{e}^{\mathrm{E}}$ is the error function representing the Euclidean distance between $\mathbf{x}_v^t$ and $\mathbf{x}_v^s$ and is defined by:

$$\mathbf{e}_v^{\mathrm{E}}(\boldsymbol{\xi}) = \|\mathbf{x}_v^t - \mathbf{x}_v^s(\mathbf{f}^{meas}, \boldsymbol{\xi})\| \tag{5.6}$$

$\mathbf{x}_v^t$ is taken at the final deformed state and $\mathbf{x}_v^s$ at the end of the estimation simulation. The estimated parameters are the ones for which the cost-function S is minimized using the Levenberg-Marquardt algorithm [Mor78]

The simulation applies a force field, which replicates the forces that were recorded during the experiment, on a region whose barycenter matches the contact point location on the simulated object and whose area is equivalent to the maximum area of contact during the experiment . The Jacobian matrix of the cost-function is needed to compute the update of the elasticity parameters between two steps k and $k+1$ of the minimization algorithm. This Jacobian matrix is numerically estimated by running several corotational FEM simulations using different values of elasticity parameters and computing the error between the experimental data and the simulation result. Let $\mathbf{J}^{\mathrm{E}}$ be the Jacobian matrix at this step of the function $\mathbf{e}^{\mathrm{E}}$with regard to the elasticity parameter vector $\boldsymbol{\xi}$, $\Delta\boldsymbol{\xi}(k)$ be the update of the elasticity parameter vector at the k^{th} step and μ be a damping factor that is adjusted at each Levenberg-Marquardt step. The update of the elasticity parameter vector can be computed by solving for $\Delta\boldsymbol{\xi}(k)$ the linear equation:

$$\Delta\boldsymbol{\xi}(k) = \Big((\mathbf{J}^{\mathrm{E}})^\top \mathbf{J}^{\mathrm{E}} + \mu \mathrm{diag}\Big((\mathbf{J}^{\mathrm{E}})^\top \mathbf{J}^{\mathrm{E}}\Big)\Big)^{-1} \Big((\mathbf{J}^{\mathrm{E}})^\top \mathbf{e}^{\mathrm{E}}\Big) \tag{5.7}$$

$$\boldsymbol{\xi}(k+1) = \boldsymbol{\xi}(k) - \Delta\boldsymbol{\xi}(k) \tag{5.8}$$

During the experimental evaluation described in Section 5.3, we focused on estimating the Young modulus E The Poisson's ratio is assumed to be known a priori. Thus, the elasticity parameter vector is given by $\boldsymbol{\xi}= (E) \in \mathbb{R}$. At this stage, the method permits to estimate the Young modulus $\hat{E}$ of a deformable object from deformation observations and external force measurements.

5.2.3 Remote force estimation

When doing *remote force estimation*, deformation tracking can be performed without external force measurements, as illustrated in Figure 5.2.

Let A be the set of vertices representing the active region of the object. The active region is an approximation of the surface of the model where the deformation occurs. This region is determined using some practical heuristics. For instance, neither the vertices lying on the ground plane nor the vertices not visible from the camera are taken into account. Active vertices, denoted by a, are the vertices belonging to the active region. The external deformation forces of the vertices belonging to the active region $\widehat{\mathbf{f}_a^{ext}}$ are estimated thanks to the physics-based simulation used for the deformation tracking. However, these forces have no physical meaning as the real object is not actually segmented as the mesh is. Instead, we consider the resultant of these forces, which corresponds to the external deformation forces applied onto the global object and is given by:

$$\widehat{\mathbf{f}^{ext}} = \sum_a \widehat{\mathbf{f}_a^{ext}} \tag{5.9}$$

The principle of the remote force estimation is depicted in Figure 5.5.

The estimated forces are projected onto the normal of the active surface $\mathbf{n}^A = \frac{\sum_a \mathbf{n}_a}{\|\sum_a \mathbf{n}_a\|}$ because the tracking method cannot have information about tangential forces, since Equation (5.3) is a point-to-plane distance. This produces the following estimate of the magnitude of the deformation forces:

$$\|\widehat{\mathbf{f}^{ext}}\| \approx \widehat{\mathbf{f}^{ext}} \cdot \mathbf{n}^A \tag{5.10}$$

5.3 Experiments

5.3.1 Experimental setup

Our experimental setup shown in Figure 5.6 consists of a 6-DOF anthropomorphic robot arm (a Viper 850 from ADEPT) equipped with a ATI's Gamma IP65 force-torque sensor and a 3D-printed stylus used as an end-effector distal tool. A RGB-D Intel Realsense D435 camera is used for the tracking. Our method has been implemented on a computer equipped with an Intel Xeon CPU running at 3.70GHz with 16 logical cores. Our method has been tested with different soft objects: a foam block, a soft ball and a complex-shaped plush toy.

The transformation between the robot frame and the object frame is computed once at the initialization of our method. To determine the constant transformation between the remote

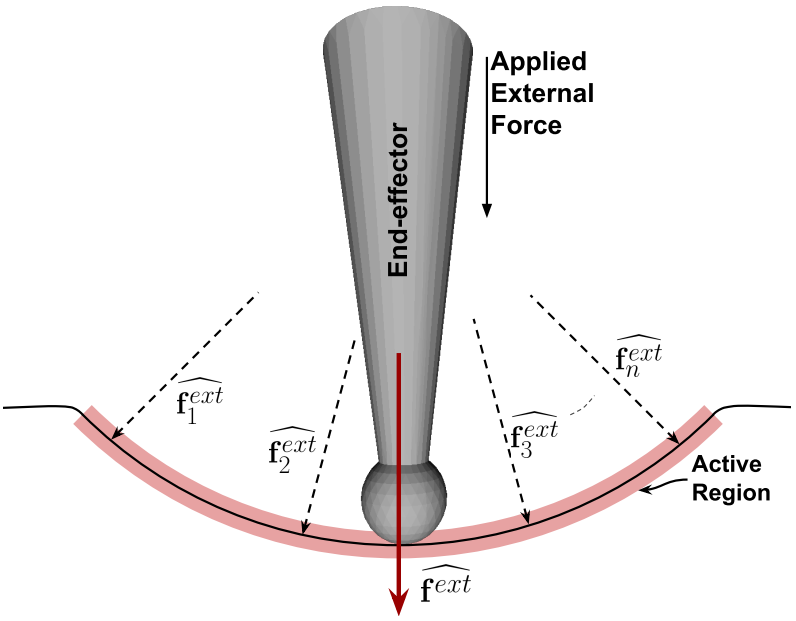


Figure 5.5: Force estimation in the active region.

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 and camera frames. The tracking of the robot end-effector uses a tag made of four circles. The circles are tracked through the RGB video stream using ViSP library [MSC05]. The initial homogeneous transform between the object and the camera frame is obtained in the first image frame using pre-trained markers [MUS16] . We finally get the transformation between the robot frame and the object frame by chaining the constant transformation between the remote camera frame R_c and the robot base frame R_r and the initial homogeneous transform between the object and the camera frames.

We then chain the transformation between the robot frame and the object frame 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. Thereafter, it is used to express measured forces into the object Cartesian frame.

The mesh of the plush toy has been generated by photogrammetry using the Meshroom software [MMM12]; [JP11]. The corotational FEM simulations are performed using SOFA framework [Fau+12].

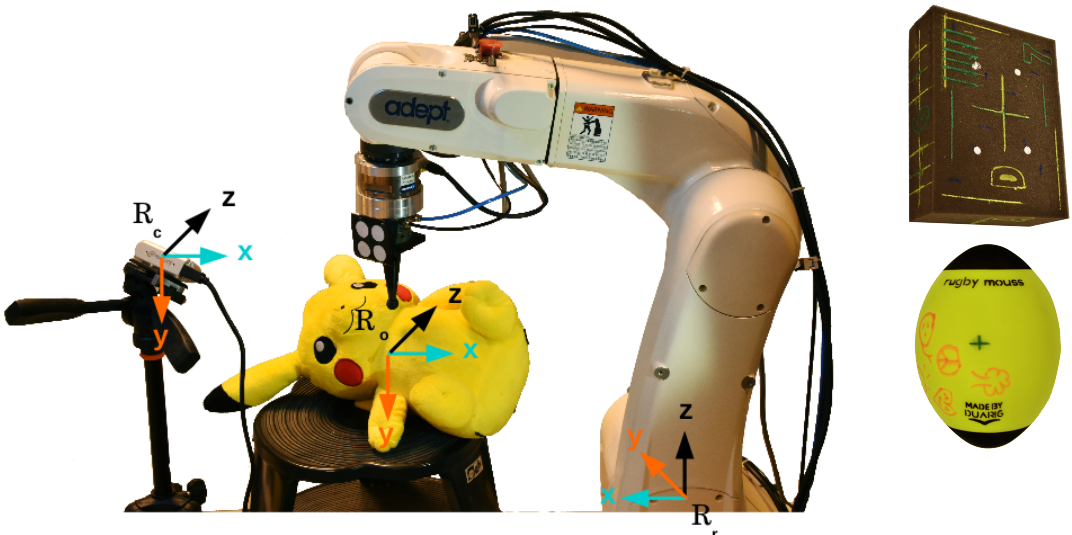


Figure 5.6: (Left) Experimental setup showing the robot (R_r) deforming a plush toy (R_o) while the data is captured from an RGB-D Camera (R_c.) The robot end-effector is equipped with a force-torque sensor. (Right) The other objects that have been experimented upon.

5.3.2 Results

The following section describes a series of experimental conditions and their corresponding results. Each *condition* represents a particular position of a specific object, e.g., the first *condition* describes the rectangular foam being deformed at the center of its largest face, while being placed horizontally on a table. To evaluate the accuracy of the estimation process, the ground truth of Young modulus E_{GT} has been determined through indentation tests for each condition of experiment [McK+11]. The indentation tests are repeated, slow and incremental vertical displacements applied to the objects using the robot while measuring the forces and the displacements. Forces are measured from the force sensor while displacements are obtained using the odometry of the robot. For each condition of experiment, the average of several indentation tests is taken as ground truth $\overline{E_{GT}}$.

Let $\hat{E}(i)$ be the Young modulus estimated using the i^{th} deformation tracking results. The Young modulus that is thereafter used for the external force estimation, denoted by E_{est}, is the one for which the following convergence criterion is respected:

$$\frac{\hat{E}(i) - \hat{E}(i-1)}{\hat{E}(i-1)} < 0.05 \tag{5.11}$$

Some examples of the output of the deformation tracking method are shown in Figure 5.7. The mean of the norm of the error vector ($\overline{\|\mathbf{e}^N\|}$) was found to be 1.53 mm, 0.51 mm and 0.21

mm for the foam block, soft ball and plush toy, while the standard deviations were 7.2 mm, 0.7 mm and 1.4 mm respectively, across all the experiments reported in this article. In the three example sequences shown in Figure 5.7, the mean of the norm of the error vector varied between a maximum of 0.381 mm (for the plush toy, deformed by its nose) to as low as 3 μm (for the undeformed sponge block), despite having outlying correspondence or noise in the range of -5.5 cm to +9.0 cm (which gets rejected by the M-estimator).

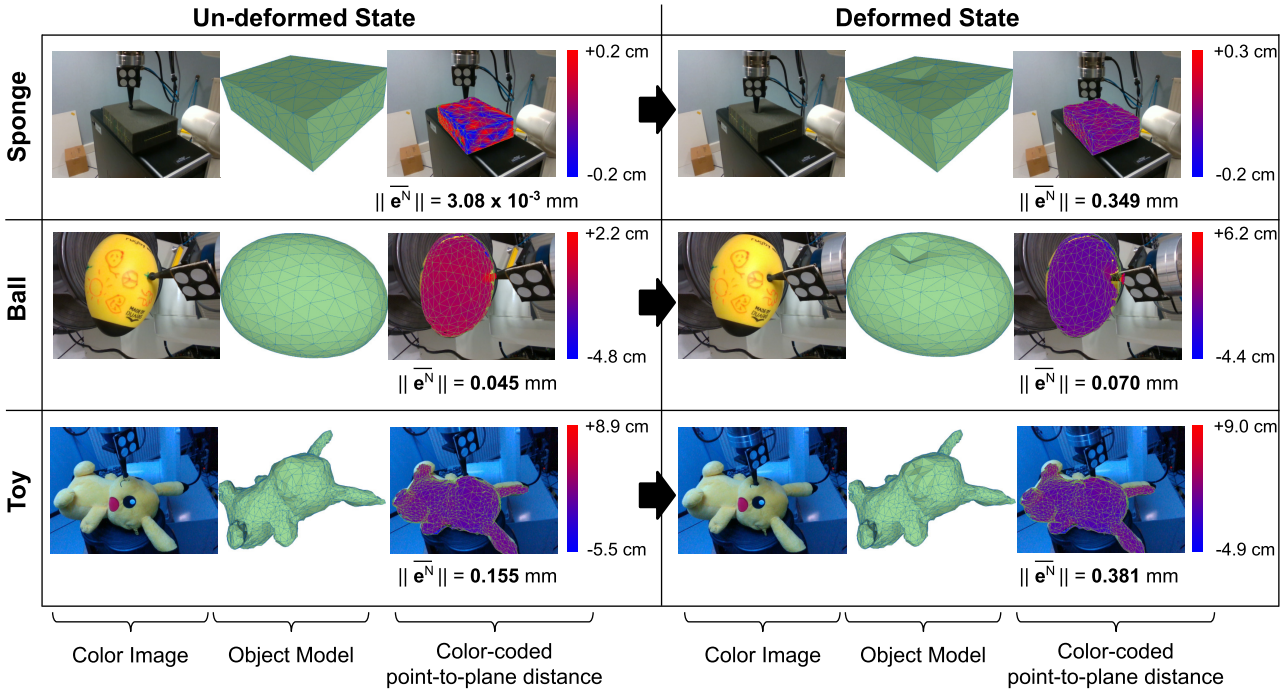


Figure 5.7: Tracking of the objects to estimate their elasticity parameters. The first and fourth columns show the color image, the second and fifth columns show the tracked object model and the third and sixth columns show the value of $\mathbf{e}^N$, color-coded and augmented on the object.

In the experiments shown in Table 5.1, the time required for tracking varies between 1.21 to 3.1 sec/frame. However, it was possible to run the same algorithm on the foam block at $\sim$ 800 ms/frame without using any GPU based computation, with negligible loss of accuracy ($< 10\%$).

Our method can be evaluated on different criteria. The first set of experiments was carried out to assess the time-performance of the tracking and the quality of the Young modulus estimation with regard to the number of vertices of both the visual mesh and the mechanical mesh. The results of this series of experiments are collated in Table 5.1. In this table, #Visual designates the number of vertices of the visual mesh while #Mechanical designates the number

Table 5.1: Evaluation of tracking and estimation times and Young modulus (in kPa) estimation accuracy with regard to different quality of visual and mechanical meshes.

Objects	**#Visual**	**#Mechanical**	$\overline{t_{tr}}$ **(s)**	**std**(t_{tr}) **(s)**	t_{st} **(s)**	t_{est} **(s)**	$\overline{E_{GT}}$	E_{est}	**Error (%)**
foam	35	1049	1.94	0.14	62	310	454	431	5.08
foam	415	1049	2.37	0.15	122	180	454	438	3.62
foam	178	554	1.21	0.02	60	180	454	497	9.47
ball	404	627	1.38	0.03	126	70	156	136	12.8
ball	404	1060	1.61	0.02	118	96	156	149	5.0
ball	404	1954	3.09	0.07	130	330	156	148	5.1

of vertices of the mechanical mesh. $\overline{t_{tr}}$ is the average deformation tracking time, std(t_{tr}) is its standard deviation, t_{st} is the time required to deform the object and reach a steady-state and t_{est} is the time to estimate the Young modulus. All the times are expressed in seconds. $\overline{E_{GT}}$ and E_{est} correspond respectively to the average ground truth of Young modulus and the final estimated Young modulus, both expressed in kilo Pascals. Finally, Error designates the percentage of error of the estimation and is given by $Error = 100|\overline{E_{GT}} - E_{est}|/\overline{E_{GT}}$.

The second series of experiments, whose results are collated in Table 5.2, was conducted to evaluate the consistency of the Young modulus estimation with regard to the initial estimate E_0. Experiments were conducted on two locations of the foam and the plush toy, as indicated in the first column of Table 5.2, to evaluate if the method can handle non-homogeneous objects. Low initial estimate are of the order of $0.1 \times \overline{E_{GT}}$ and high initial estimate of the order of $10 \times \overline{E_{GT}}$.

The last series of experiments was conducted to evaluate the estimation of the deformation forces under different experimental conditions. For these experiments, the estimated Young modulus values that are summarized in Table 5.2 were used by the deformation tracking method. The results of this series of experiments are collated in Table 5.3.

$\overline{f_{GT}}$ and f_{est} designate respectively the average norm of the ground truth forces measured by the force sensor and the norm of the forces estimated by the deformation tracking method, expressed in Newtons. $\Delta f = f_{est} - \overline{f_{GT}}$ represents the error between the estimated and ground truth forces. The estimated forces are projected onto the measured forces. The force experiments were conducted with different orientations of the objects (horizontal and tilted approximately 25 degrees from the Z-axis of the object frame). Different depths of stimulation were

Table 5.2: Evaluation of the estimation of the Young modulus (in kPa).

Objects	E_0	$\overline{E_{GT}}$	E_{est}	**Error**
foam middle	200	454	480	5.7%
	1500	454	451	0.7%
	15000	454	488	7.5%
foam corner	200	247	224	9.3%
ball	1000	156	154	1.3%
toy nose	200	51	53	3.7%
toy leg	50	42	43.1	2.6%

Table 5.3: Comparison of the average norm of the measured and estimated forces.

Objects	**Orientation**	**Depth (cm)**	$\overline{f_{GT}}$ **(N)**	f_{est} **(N)**	Δf **(N)**
foam	horizontal	2	18.8	20.6	1.8
foam	tilted	2	21.36	23.11	1.75
foam	horizontal	4	45.46	42.67	-2.79
foam	tilted	4	38.94	35.59	-3.35
ball	horizontal	2.5	9.49	8.57	-0.92
toy	horizontal	2	5.32	5.81	0.49

also tested and are indicated in the depth column of Table 5.3.

Finally, Figure 5.8 depicts a possible use case of our method to estimate deformation forces exerted on an object in an environment where no force measurements are available using the method proposed in Section 5.2.3. We use the Young modulus that was estimated during the experiments summarized in Table 5.2. A human operator exerts a pressure on the nose of the plush toy. By deforming the object deeper than using the end-effector, the estimated deformation forces are of 13.8N, and are within the expected range.

5.4 Discussion

In this chapter, we proposed a method to simultaneously track arbitrary linear elastic objects and estimate their elasticity parameters. To estimate the elasticity parameters, our method

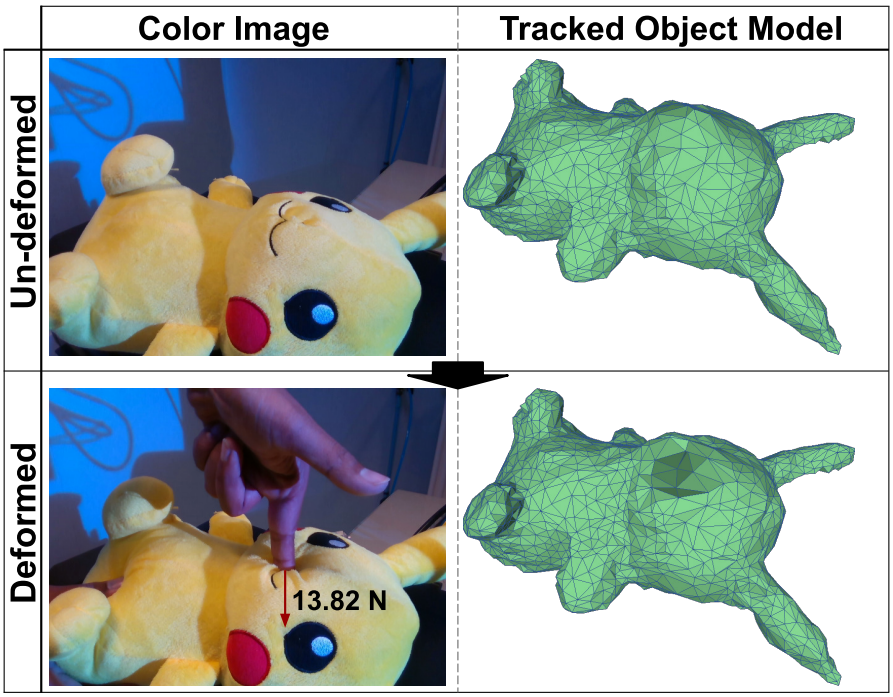


Figure 5.8: Force estimation while the plush toy gets freely deformed by hand. The deformation forces (13.8 N) are in the expected range.

only requires a coarse 3D mesh of the object and external force measurements. The first series of experiments whose results are summarized in Table 5.1 showed that the accuracy of the Young modulus estimation is improved as the resolution of the mechanical mesh increases. This results from the fact that the deformation can be represented with finer details and is thus closer to reality. Table 5.2 shows that our method is able to estimate the Young modulus of an arbitrary object even when the initial estimate is much different from the ground truth. Table 5.3 shows that the estimated Young modulus can be used by our method to accurately estimate the deformation forces from visual information only. The use case scenario depicted in Figure 5.8 shows a possible application of our method in environments where no force sensor could be used, for instance in human-performed manipulation tasks. This application opens novel perspectives for better estimating the forces applied to a deformable object by human operators, as well as on a cooperative task on such objects.

Chapter 6

Conclusion

This manuscript discussed the topic of the dexterous manipulation of deformable objects by robots and presented our contributions to this topic. It particularly focused on the shape control of deformable objects and on estimating some of their elasticity parameters.

One of the challenge arising from the dexterous robotic manipulation of non-rigid objects is to predict their deformation behavior. This challenge combines the fact of establishing deformation models and developing model-free deformation methods. Our contribution to address this challenge was a model-free deformation servoing method that enables the robots to understand the relation between their actions and the deformation that the non-rigid object is undergoing while not relying on physical models.

A second challenge arising from the dexterous robotic manipulation of non-rigid objects is to manipulate non-rigid objects while controlling their shape. To address this challenge in the context of linear deformable objects, we proposed a novel method that permits dual-arm manipulation of deformable wires without using any physical model. Thanks to this method, the shape of deformable wires can be controlled at an interactive time.

Finally, another challenge arising from the dexterous robotic manipulation of non-rigid object is to determine the object physical parameters while avoiding damaging the object. To address this challenge, we proposed a method to simultaneously track a deformable object and estimate some of its elasticity parameters.

These works pave the way for improving the dexterity and the autonomy of robots interacting with unknown deformable objects.

6.1 Active deformation through visual servoing of soft objects

In Chapter 3, we proposed the ADVISEd method for autonomous shape control of deformable objects by robots. The method does not depend on any model of the object or its interactions

with the environment. Instead, a deformation Jacobian matrix relating the motions of the end-effector(s) to the object deformations is estimated online. The estimation is performed by a weighted least-squares minimization with a sliding window. To increase the robustness of the estimation to noise measurement, we propose to update the deformation Jacobian using a confidence criterion. The method has been validated by comparisons with state-of-the-art methods. The evaluation has been conducted using two different experimental setups: one for a marker-based active shaping task and one for multiple marker-less active shaping and shape preservation tasks. Experiments have shown that the method permitted to interactively control the deformations of a soft object while ensuring a better robustness to external perturbations. Experiments also showed that the method can be used to generate large deformations by subdividing the task into intermediary tasks.

6.2 Automatic shape control of deformable wires based on model-free visual servoing

In Chapter 4, we proposed a method to dexterously manipulate deformable wires using a dual-arm robot. To permit autonomous shape control of wires, we proposed a novel deformation feature that relies on a B-spline geometric model and permits to model the 3D shape of a deformable wire. We also proposed a marker-less model-free tracking method based on a Sequential Importance Resampling particule filter that permits to track this feature at an interactive time. This novel deformation feature is used to compute a control law using a model-free shape servoing method. The deformation Jacobian matrix used to compute this control law is estimated online. Our method was tested on several wires with different materials, and similar accuracy was obtained across all tested wires. Experiments have also shown that the method can be used to reach complex deformed 3D shapes by defining intermediary targets.

6.3 Simultaneous tracking and elasticity parameter estimation of deformable objects

In Chapter 5, we proposed an iterative method to simultaneously track and estimate some of the elasticity parameters of a deformable object. First, the object is monitored by an **RGB-D** sensor while being probed with an end-effector equipped with a force-torque sensor. When the object is in a steady-state, the deformation tracking results are used along with the force

measurements in a minimization process relying on physics-based simulation. The results of the elasticity parameter estimation are then used in the deformation tracking method, thus closing the loop. Once the elasticity parameter estimation has converged, the deformation tracking can estimate deforming contact forces from vision data solely. Experiments have shown that the method can be used to track and estimate the Young modulus of objects of different geometry and materials. Once the elasticity parameter estimation has converged, remote force estimation could be performed on a complex object.

6.4 Short-term perspective

6.4.1 Active deformation through visual servoing of soft objects

An interesting future work would be to use the ADVISEd method presented in Chapter 3 with different deformation features and/or tracking methods. For instance, the use of Scale Invariant Feature Tracking would permit to have a continuous registration between the **RGB-D** sensor and the deformation feature. This would permit to get rid of the hypothesis that prevented the object from moving relatively to the camera and thus ease the deployment of the method in uncontrolled environments. The ADVISEd method could also be used with a planner to achieve complex tasks such as pick-and-place of deformable objects. In such a task, the ADVISEd method could be used to manage the shape preservation of the deformable object while the planner would manage the positioning task.

6.4.2 Automatic shape control of deformable wires based on model-free visual servoing

An interesting future work for the automatic shape control of deformable wires method presented in Chapter 4 could be to automatically generate intermediary targets from initial and desired shapes. The automatic generation of intermediary targets would ease deployment in autonomous systems. This could be done using for instance data-based methods, during a step prior to the interaction so as not to impact the interactive time abilities of the proposed method. Additionally, by coupling this method to a planner able to estimate the affordance of actions, it could be possible to achieve complex tasks such as knotting. Knotting tasks bring additional challenges such as occlusion of the deformable wire or the need to swap the two extremities of the wire between the end-effectors. These challenges could, for example, be addressed by

detecting when the two end-effectors would intersect along the way relatively to the **RGB-D** sensor, adapting therefore how the end-effectors hold the wire.

6.4.3 Simultaneous tracking and elasticity parameter estimation of deformable objects

A possible future work for the STEPE method presented in Chapter 5 would consist in estimating local friction parameters with the Young modulus by considering the dynamics in the parameter estimation process. It would come handful for the automatic grasping of non-rigid objects, thus broadening the range of possible interactions for autonomous systems. Another future work would consist in deriving an analytic expression of the Jacobian used in the estimation process in order to accelerate the estimation process, removing the need for additional simulations. It would allow autonomous systems to adapt their behavior more quickly to their environment, which could be critical for time-sensitive tasks.

6.5 Long-term perspective

6.5.1 Towards adaptive humanoid robots

As of today, works on humanoid robots are in progress as they could be of great help with tasks requiring a lot of strength or for caregiving. When the technology will be mature, humanoid robots will have to interact with deformable objects in the daily life. Giving to humanoid robots the ability to explore deformable objects without damaging them will be of critical interest to enable them to extend their knowledge database online.

The work presented in this manuscript is a first step towards adaptive robots. Model-free shape servoing methods could be used as a tool in more complex systems to permit robots to manipulate unknown objects while ensuring their integrity. Used with a parameter estimation method, a model-free shape servoing method could be used as initial phase for model-based methods when dealing with new types of deformable objects. Such a combination would permit humanoid caregiver robots to help people in their daily life while ensuring their safety.

6.5.2 Towards adaptive arm prostheses and arm exoskeletons

The purpose of arm prostheses is to help who have lost one of their arms. Finding an intuitive way to control these prostheses is of critical interest to make the help they give the best possible.

Arm exoskeletons are external devices that permit for instance to lighten the effort that one must provide to accomplish a physical task. It is interesting for instance in factories where operators have to repeat physical tasks all day long.

Model-free shape servoing methods can be useful for both arm prostheses and arm exoskeletons control. Indeed, it could permit the user not to have to focus on the force to exert when holding a non-rigid object. To do so, the shape servoing method should use cues of different nature from the vision, such as force or tactile measurements. Additionally, if the shape servoing method is used along with a parameter estimation method, adding **Augmented Reality (RA)** feedbacks such as warning when the object is approaching its breaking point might improve user experience and prove handful for daily life uses.

The dexterous manipulation of deformable objects by robots is a field of research that has a bright future. Hardware improvements will ease the development of new methods through new computing capabilities and new sensors that will be available at affordable prices. Consequently, the field of applications has yet to be defined as every day brings reflections on new possible applications.

Appendix A

Appendix: Résumé long en français

A.1 Introduction

Dans ce manuscrit de thèse, intitulé *Shape control of deformable objects by adaptive visual servoing*, nous présentons des travaux de recherches portant sur la manipulation par des robots d'objets déformables.

Le travail effectué dans le cadre de cette thèse a été financé par le projet européen Imagine, pour lequel travaillent en collaboration des universités de plusieurs pays ainsi qu'une entreprise privée. Le projet Imagine vise à relever des défis découlant du recyclage automatisé d'appareils électroniques, qui peuvent contenir de nombreux composants déformables tels que des câbles, en utilisant des robots. Le recyclage des appareils électroniques devient un enjeu majeur car les matières premières nécessaires pour leur fabrication existent en quantité limitée. La manipulation d'objets déformables par des robots est une tâche non-triviale, car les objets se déforment lorsqu'ils sont manipulés. Développer des méthodes permettant aux robots de manipuler des objets déformables contribuerait grandement à automatiser le recyclage, permettant de préserver la santé des opérateurs humains qui doivent faire face aux composés chimiques présents dans les objets électroniques. C'est dans ce contexte que s'inscrit les travaux effectués au cours de cette thèse. Ces travaux ont pour but de proposer de nouvelles méthodes permettant de contrôler des robots pour qu'ils puissent manipuler des objets déformables. La manipulation d'objets déformables par des robots fait émerger des défis scientifiques tels que la prédiction du comportement de déformations d'objets déformables, la détermination de paramètres physiques d'objets déformables ou encore le contrôle de leur forme tout au long de leur manipulation.

Prédiction du comportement de déformations d'objets déformables

L'un des défis scientifiques découlant de la manipulation robotisée d'objets déformables est de pouvoir prédire les déformations que subissent ces objets lorsqu'ils sont sujets à des perturbations extérieures. Par exemple, pour pouvoir saisir un objet non-rigide, il faut être à même de

prédire la force qu'il est possible d'exercer sur celui-ci sans qu'il ne soit endommagé de façon irréversible. Il faut cependant que la force exercée soit suffisante pour qu'il ne glisse pas lors de la saisie. Pour faire cela, il est possible d'utiliser des méthodes dites *basées modèles*, c'est-à-dire utilisant un modèle physique de l'objet et/ou de ses interactions avec son environnement. Les méthodes basées modèles ont l'avantage de permettre de manipuler les objets déformables avec une grande précision, mais ont l'inconvénient d'être très sensibles à la précision de l'estimation des paramètres physiques de l'objet et d'être difficiles à transposer d'une situation à une autre.

Pour manipuler des objets déformables, une alternative au fait de *prédire* leur comportement de déformations est de donner aux robots la possibilité de déterminer à la volée les effets de leurs actions sur l'objet non-rigide avec qui ils interagissent. Pour cela, il faut fournir aux robots la possibilité de suivre la déformation de l'objet en temps quasi réel. La relation entre les actions du robot et la déformation résultante doit être déterminé en un temps suffisamment court pour pouvoir déterminer les prochaines actions à effectuer dans un délai assurant l'intégrité de l'objet. Les méthodes utilisant ce procédé sont appelées *méthodes non-basées modèles*. Elles ont l'avantage de se transposer facilement d'une situation à une autre, mais leur précision doit être vérifiée.

Détermination de paramètres physiques d'objets déformables

Un second défi émergeant de la manipulation robotisée d'objets déformables est l'établissement de procédures permettant de déterminer des paramètres physiques de ce type d'objets. Les paramètres physiques de l'objet à déterminer dépendent des types d'interactions pouvant résulter des manipulations souhaitées. En effet, selon les types d'interactions, un modèle plus ou moins complexe de l'objet sera nécessaire. La détermination des paramètres physiques nécessite d'effectuer un grand nombre de manipulations de l'objet auquel on s'intéresse tout en assurant son intégrité afin de récolter un nombre suffisant de données. Au cours de ces manipulations, il est nécessaire de pouvoir suivre de façon précise les déformations que subissent l'objet sans pour autant que les performances temporelles ne soient impactées.

Contrôle de la forme d'objets déformables

Un troisième défi émergeant de la manipulation robotisée d'objets déformables est de permettre un contrôle précis de la forme de l'objet. Le contrôle de la forme peut désigner soit le fait de conserver la forme d'origine de l'objet, ce qui s'avère nécessaire par exemple lors de tâches de saisies d'objets, soit le fait de modeler l'objet afin de lui changer sa forme. Le contrôle de la

forme d'un objet non-rigide par l'intermédiaire de robots sera désigné par le terme *déformation active* dans la suite de ce manuscrit. Les méthodes de déformations actives doivent permettre de surmonter plusieurs difficultés associées. Tout d'abord, il faut pouvoir suivre en temps réel les déformations que subit l'objet tandis qu'il est manipulé. De plus, les méthodes de déformation actives doivent être transposables d'une situation à une autre afin de ne pas avoir à concevoir un nouveau système pour faire face à une nouvelle situation.

Nos trois contributions pour relever ces défis scientifiques sont les suivantes:

- une méthode non-basée modèle permettant la déformation active d'objets déformables;
- une méthode non-basée modèle permettant de suivre en temps interactif des câbles déformables et de les manipuler à l'aide de deux effecteurs robotiques;
- une méthode permettant le suivi de déformation et l'estimation de paramètres d'élasticité d'objets déformables.

A.2 Déformation active d'objets déformables

Contexte introductif

Les robots sont utilisés depuis plusieurs décennies dans les usines du fait de leur efficacité à manipuler avec précision et à grande cadence des objets rigides. Cependant, une grande majorité des objets du quotidien sont déformables et se déforment voire se détériorent lorsqu'ils sont sujets à des forces extérieures. De ce fait, les robots industriels ont à ce jour de grandes difficultés à manipuler des objets déformables. Des recherches sont menées à ce jour afin de donner aux robots la possibilité d'interagir avec des objets déformables. Dans ce manuscrit, nous divisons les méthodes de l'état de l'art en deux grandes catégories: les méthodes dites *basées modèles* et les méthodes dites *non basées modèles*.

Les méthodes basées modèles utilisent des modèles physiques de l'objet et/ou de son environnement. Elles sont généralement combinées à des simulations basées physiques pour prédire les déformations de l'objet. Différents types de modèles peuvent être utilisés. Par exemple, Smolen and Patriciu proposent un modèle physique basé particules et ne nécessitant pas de maillage pour représenter l'objet [SP09]. A la place de modéliser uniquement l'objet non-rigide, il est possible de modéliser également les contacts entre l'objet et l'effecteur robotique. Navarro-Alarcon *et al.* proposent une méthode utilisant ce type de modélisation, les contacts étant assimilés à des systèmes masse-ressort [Nav+14]. Les mêmes auteurs ont étendu leurs

travaux sur le contrôle robotique de déformations en proposant une méthode permettant simultanément de positionner et manipuler la forme d'un objet déformable [Nav+16]. D'autres approches utilisent des modèles masse-ressort qui permettent d'obtenir une bonne vitesse de calculs. Cependant, les modèles masse-ressort n'utilisent pas directement des grandeurs physiques associées aux propriétés rhéologiques des objets et peuvent de ce fait avoir un comportement peu réaliste. Afin d'obtenir des simulations physiques de déformations plus précises, il est possible d'utiliser la Méthode des Eléments Finis (*MEF*). C'est par exemple le choix effectué par Zhang *et al.* pour contrôler des robots déformables [Zha+17]. La simulation basée physique utilisant la *MEF* a aussi été utilisée dans le cadre d'insertion automatique d'aiguilles flexibles dans des tissus vivants dans le travail d'Adagolodjo *et al.* [Ada+16]. Les méthodes basées modèles se révèlent très précises pour prédire les déformations d'objets déformables tant que les propriétés physiques et géométriques de l'objet sont connues. Cependant, à ce jour l'acquisition de ces propriétés reste très chronophage et les méthodes basées modèles ne parviennent pas à obtenir un niveau de précision sur des systèmes réels équivalent au niveau obtenu dans les simulations.

Parmi les méthodes non basées modèles de l'état de l'art, il est possible par exemple d'évoquer celle proposée par Berenson qui permet de déformer localement des objets linéaires ou planaires [Ber13]. Alonso-Mora *et al.* ont quant à eux proposé une méthode permettant de simultanément positionner et préserver la forme d'objets planaires en utilisant plusieurs robots mobiles [Alo+15]. Cependant, la préservation de la forme est une tâche secondaire qui est négligée dans le cas où elle engendrerait une collision. Navarro-Alarcon *et al.* ont proposé une méthode permettant de déformer des objets déformables en estimant en ligne la matrice Jacobienne de déformation [Nav+13]. L'estimation se base sur l'algorithme de Broyden qui estime le changement de la matrice Jacobienne à partir d'observations passées et courantes [Bro65]. Cependant, l'algorithme de Broyden est très sensible au bruit : lorsque le système converge vers la configuration désirée, certaines directions auxquelles est liée la matrice Jacobienne ne sont plus stimulées, entraînant une mise à jour uniquement à partir du bruit de mesure. Par la suite, Navarro-Alarcon *et al.* ont proposé une autre méthode qui utilise une transformation en série de Fourier des contours 2D de l'objet déformable pour réaliser la déformation active [NL18]. Cette méthode tombe dans un minimum local si le contour désiré n'est pas atteignable ou si le nombre de coefficients de Fourier utilisés pour décrire la forme de l'objet n'est pas suffisamment conséquent.

Nos contributions à la déformation active d'objets déformables, publiées dans [LKM20a], sont les suivantes:

- la méthode *ADVISEd*, qui est une nouvelle méthode non basée modèle;

- deux méthodes d'évaluation permettant de comparer la méthode ADVISEd à des méthodes de l'état de l'art.

ADVISEd, une méthode de déformation active d'objets déformables

La méthode ADIVSEd est une méthode de déformation active d'objets déformables non-basée modèle prenant inspiration sur les principes de base des méthodes d'asservissement visuel. Le principe est d'utiliser l'erreur entre les caractéristiques visuelles de déformation actuelles et celles désirées pour calculer la loi de commande permettant de contrôler en vitesse le ou les effecteur(s) robotiques. La variation des caractéristiques visuelles de déformation est liée à la variation de la vitesse du ou des effecteurs grâce à la matrice Jacobienne de déformation. Nous proposons d'estimer en ligne cette matrice Jacobienne de déformation en utilisant une méthode de minimisation des moindres carrés avec fenêtre glissante. La fonction de coût minimisée est une somme des carrés des différences entre les variations des caractéristiques visuelles observées et celles qui sont calculées en utilisant la matrice Jacobienne et les vitesses des effecteurs. Pour améliorer la robustesse de la mise à jour de la matrice Jacobienne aux bruits de mesure, nous proposons d'ajouter un critère de confiance qui repose sur une décomposition en valeurs singulières de la matrice.

Une fois la matrice Jacobienne de déformation mise à jour, elle est utilisée dans une loi de commande en boucle fermée illustrée dans la Figure A.1 afin de contrôler le ou les effecteur(s) pour déformer l'objet.

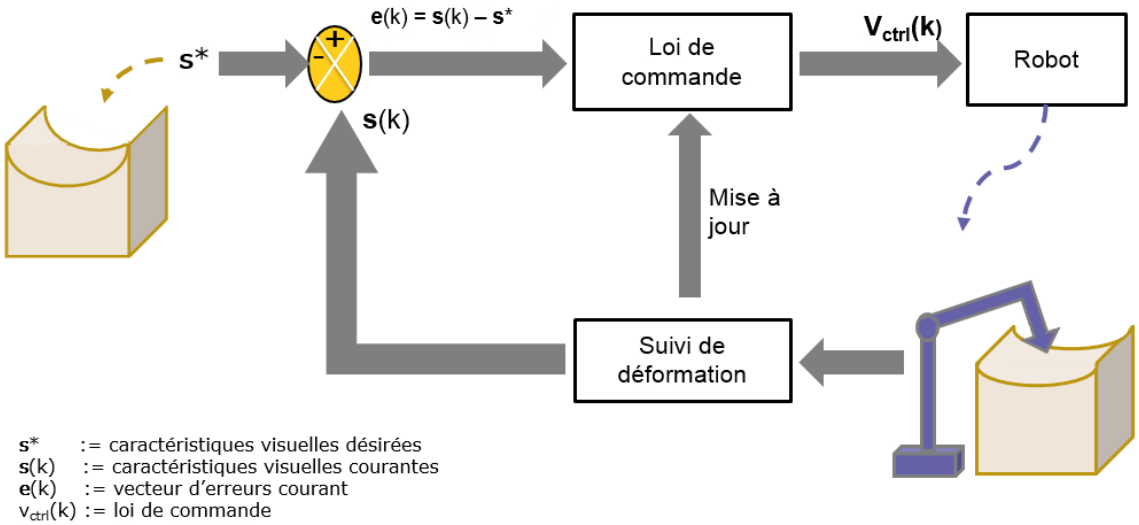


Figure A.1: Schéma de principe de la méthode ADVISEd.

Nous proposons deux méthodes d'évaluation de la méthode ADVISEd : une reposant sur des marqueurs visuels et une ne reposant pas sur des marqueurs. Dans la méthode reposant sur des marqueurs, les caractéristiques visuelles de déformation utilisées sont les coordonnées 3D des marqueurs exprimées dans le repère caméra.

La méthode d'évaluation ne reposant pas sur des marqueurs utilise les coordonnées 3D d'une surface d'intérêt délimitée sur l'objet en guise de caractéristiques visuelles de déformation. Pour réduire le bruit de mesure, une méthode de M-estimateur est utilisée et modifie légèrement la formulation de la loi de commande.

Évaluation de la méthode ADVISEd

Lors de l'évaluation de la méthode ADVISEd en utilisant des marqueurs visuels, les critères d'évaluation utilisés sont : la norme euclidienne du vecteur d'erreur à la fin de la manipulation, le temps de réponse du système, correspondant au temps nécessaire pour réduire l'erreur à 10% de sa valeur initiale, le temps de convergence, correspondant au temps à partir duquel l'erreur reste dans un intervalle de plus ou moins 5% la valeur médiane finale, et la distance totale parcourue par l'effecteur robotique. La norme euclidienne du vecteur d'erreur à la fin de la manipulation permet d'évaluer la précision du système. Le temps de réponse du système permet d'évaluer la réactivité du système. Le temps de convergence et la distance totale parcourue par l'effecteur robotique permettent d'évaluer la stabilité du système. La méthode ADVISEd a notamment été évaluée sur le fait de générer de grandes déformations. Les expériences ont montré qu'en subdivisant la tâche en tâches intermédiaires, la méthode ADVISEd était capable d'accomplir ce genre de tâche avec une bonne précision. Les performances de la méthode ADVISEd ont aussi été comparées à celles de méthodes de l'état de l'art: une basée modèle et l'autre non. Au cours de cette évaluation, la méthode ADVISEd s'est révélée plus précise que la méthode basée modèle et plus stable que la méthode non-basée modèle.

Lors de l'évaluation de la méthode ADVISEd sans utiliser de marqueurs visuels, les critères d'évaluation utilisés sont les mêmes que dans le cas avec marqueurs, à l'exception de la fonction d'erreur utilisée. A la place, nous considérons la distance de Hausdorff qui représente la plus grande distance euclidienne entre les points de la surface finale mis en correspondance avec ceux de la surface désirée. De plus, seule la méthode non-basée modèle de l'état de l'art a été utilisée pour la comparaison, la méthode basée modèle s'étant révélée moins précise lors des expériences précédentes. Quatre types de tâches ont été réalisées : deux tâches de déformation active, une tâche de préservation de forme et une tâche de déformation active avec perturbations extérieures. Lors de ces expériences, la méthode ADVISEd s'est montrée aussi précise que la

méthode de l'état de l'art tout en étant plus robuste aux perturbations extérieures.

A.3 Manipulation précise de câbles déformables

Contexte introductif

Les câbles déformables sont utilisés dans de nombreux domaines. De ce fait, le contrôle de leur forme peut être utile dans diverses applications, telles qu'en médecine pour l'insertion automatique d'aiguilles déformables [CKN15] ou le contrôle de cathéter [Beh+19]; [KH11]; [Fan+19] ou encore dans l'industrie pour permettre le câblage automatique[Zhu+19]. Cependant, à ce jour les robots industriels ne peuvent pas contrôler facilement la forme de câbles déformables du fait du comportement complexe de ces derniers lorsqu'ils sont sujets à des déformations. Pour contrôler la forme d'un câble déformable, il faut pouvoir suivre à une vitesse quasi-temps réel la forme 3D de l'objet. Il faut aussi pouvoir utiliser le résultat de ce suivi lors de l'élaboration de la loi de commande de robots qui sera utilisée pour contrôler la déformation de l'objet.

Différentes méthodes de suivi de forme 3D de câbles existent dans l'état de l'art. Par exemple, Matsuno *et al.* ont proposé une méthode de suivi pour le cas particulier de câbles noués [MF06]. Cette méthode est basée sur la théorie des nœuds et permet notamment d'évaluer la similarité entre deux nœuds. Plusieurs méthodes probabilistes ont été proposées. Il est possible de citer notamment les travaux de Tang *et al.* qui combinent des simulations basées physiques à une méthode probabiliste de mise en correspondance pour suivre un câble déformable dans un nuage de points [TT18]. Une autre méthode probabiliste de mise en correspondance a été proposée par Chi *et al.*, l'utilisation d'information de visibilité de l'objet permettant de ne pas tenir compte des points du maillage de l'objet n'étant pas visibles par la caméra [CB19]. Cette méthode permet aussi la reprise du suivi s'il se retrouve mis en échec sur plusieurs images successives en cherchant dans un historique une situation similaire à celle observée par la caméra. Finalement, une méthode utilisant à la fois un modèle probabiliste et un modèle physique de l'objet a été proposée par Schulman *et al.* pour permettre le suivi d'un objet déformable à partir de nuages de points [Sch+13]. Des méthodes basées sur le traitement d'images ont été proposées pour le suivi de câbles déformables. Il est notamment possible de citer les travaux d'Abegg *et al.*, où le câble est suivi dans le plan image [AW00]. Pour faire cela, l'utilisateur sélectionne au début de la séquence un point de l'image appartenant au câble. Des caractéristiques telles que la luminance ou l'orientation globale de la région sont alors extraites. Elles sont ensuite utilisées pour détecter le câble sur l'ensemble de l'image parcourue région par

région afin de trouver des régions possédant des caractéristiques identiques. Padoy *et al.* ont quant à eux proposé une méthode de suivi de câbles basée sur une augmentation de texture visuelle et un modèle géométrique appelé B-spline [PH12]. La texture visuelle du câble est augmentée en y ajoutant un motif coloré à sa surface.

Nous proposons de diviser les méthodes de contrôle de forme de câbles en deux catégories, selon qu'elles utilisent ou non des modèles mécaniques pour prédire la forme du câble ou ses interactions avec l'environnement. Parmi les méthodes basées modèle présentes dans l'état de l'art, il est possible de citer les travaux de Laranjeira *et al.* qui ont proposé une méthode d'asservissement visuel permettant de contrôler le relâchement d'un câble reliant deux robots [LDH17]. Cette méthode repose sur l'hypothèse que seule la gravité est responsable de la déformation du câble. Une méthode de contrôle de la force et de la position d'un robot de type cathéter permettant de le faire entrer en contact avec une cible mouvante a été proposée dans [KH11]. Les modèles utilisés permettent de compenser les rebonds et frictions générés par les mouvements du cathéter dans son environnement. D'autres méthodes basées modèles ont été proposées, utilisant par exemple des modèles masse-ressort comme dans les travaux publiés dans [SH02] ou la Méthode des Eléments Finis comme dans les travaux publiés dans [Din+12]. Des méthodes non-basées modèles ont été proposées, permettant d'éviter la complexité de modéliser le comportement de déformation des câbles. Par exemple, une méthode probabiliste a été utilisée dans les travaux de Hu *et al.*, qui présentent une méthode nécessitant une phase préliminaire afin de déterminer les lois de probabilités nécessaires lors de la commande [HSP18]. L'état de l'art contient aussi des méthodes basées apprentissage, telle que celle proposée par Tang *et al.* où une base de données est utilisée pour déterminer les commandes à envoyer au robot [TWT18]. Finalement, des méthodes utilisant les principes de l'asservissement visuel ont été proposées, telles que celle proposée par Zhu *et al.* qui permet de contrôler la forme 2D d'un câble en utilisant plusieurs effecteurs robotiques [Zhu+18]. La forme 2D du câble est approchée par une série de Fourier dont les coefficients sont utilisés pour calculer la loi de commande. La matrice Jacobienne de déformations utilisée dans la loi de commande est mise à jour en ligne par une méthode de minimisation des moindres carrés si le robot s'est déplacé dans toutes les directions dont il est capable.

Nos contributions pour relever des défis liés à la manipulation précise de câbles déformables sont les suivantes:

- une nouvelle caractéristique visuelle permettant d'inférer la forme 3D de câbles en utilisant un modèle géométrique;

- une méthode de suivi, basée sur du filtrage particulaire et ne nécessitant pas de marqueurs, permettant de suivre un câble à une vitesse quasi-temps réel;

- une méthode de déformation active non basée modèle permettant de contrôler deux effecteurs robotiques pour déformer un câble afin qu'il atteigne une déformation désirée.

Notre approche, publiée dans [LKM20b], ne nécessite aucune connaissance des propriétés physiques du câble pour contrôler sa forme 3D. De plus, les performances de calculs permettent de l'utiliser à une vitesse quasi-temps réel dans le cadre de différents types de manipulations impliquant éventuellement de larges déformations.

Méthodologie proposée pour la manipulation précise de câbles déformables

Le but de la méthode que nous proposons est de permettre à deux bras manipulateurs robotiques de contrôler de façon précise la forme d'un câble déformable en le tenant par ses extrémités.

Pour ce faire, nous proposons une nouvelle caractéristique visuelle permettant de décrire la forme 3D de câbles en utilisant un modèle géométrique. Le modèle géométrique que nous avons choisi est une B-spline 3D, qui est une fonction polynomiale définie par morceaux et peut être construite à partir de points de contrôle. En déplaçant les points de contrôle d'une B-spline, la forme 3D décrite par la B-spline change. Nous utilisons cette propriété pour proposer une méthode de suivi basée sur du filtrage particulaire et ne nécessitant pas de marqueurs.

Le filtrage particulaire utilise plusieurs candidats, aussi appelés particules, pour représenter l'état interne d'un système observé. Dans notre cas, les particules sont des B-splines représentant la forme 3D du câble dont on souhaite suivre les déformations. A chaque nouveau pas de temps, les particules sont mises à jour à l'aide d'une loi de prédiction. La loi de mise à jour prédiction que nous proposons , illustrée dans la Figure A.2, consiste à déplacer les points de contrôle d'un mouvement résultant de la combinaison linéaire des mouvements des bras robotiques et de bruit Gaussien. Quand toutes les particules sont mises à jour, on évalue la similitude de chacune avec le nuage de points observé par la caméra de profondeur. Cette similitude est calculée sur des fenêtres glissantes, comme illustré dans la Figure A.3. Finalement, la forme 3D du câble est obtenue comme étant la somme des particules, pondérées par leur similitude normalisée.

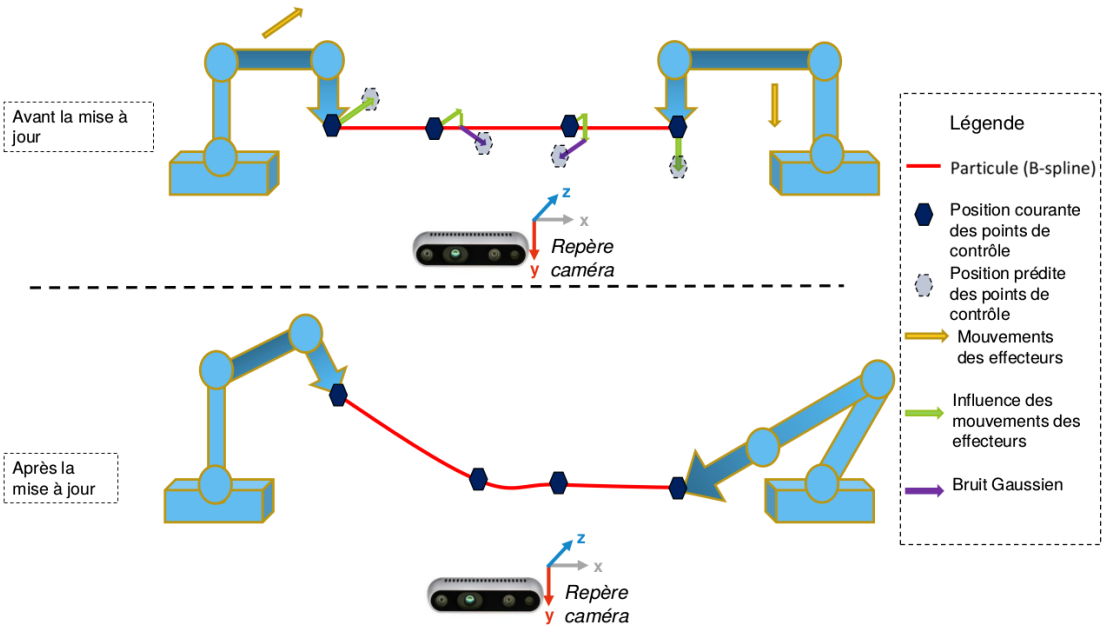


Figure A.2: Mise à jour d'une particule à partir des mouvements des effecteurs robotiques.

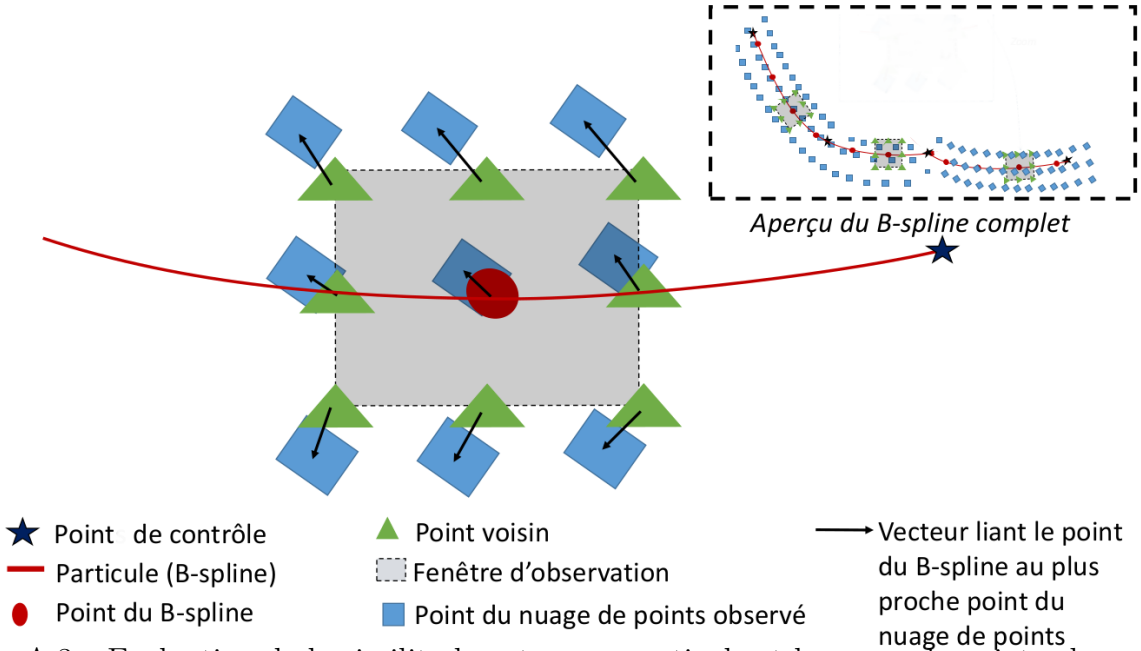


Figure A.3: Evaluation de la similitude entre une particule et le nuage de points observé par la caméra de profondeur. Dans l'encadré se trouve un aperçu de l'aspect global de la particule. Au premier plan se trouve un zoom sur l'extrémité droite de la particule.

Finalement, des points 3D équidistants en terme de paramètre curvilinéaire de la B-spline résultant de la méthode de suivi des déformations sont utilisés comme caractéristiques de dé-

formation dans la loi de commande pilotant les deux robots. L'utilisation de points 3D équidistants assure une mise en correspondance avec les points physiques du câble, permettant ainsi de décrire les déformations de type élongation que peut subir le câble. Une méthode non-basée modèle est utilisée pour mettre à jour en ligne la matrice Jacobienne de déformation utilisée dans la loi de commande.

Évaluation

La méthode de manipulation précise de câbles a été évaluée sur les critères suivants: le déplacement moyen entre le câble dans son état de repos et dans l'état déformé final, l'erreur moyenne correspondant à la distance euclidienne entre un point du câble et sa position désirée, l'erreur relative exprimant le ratio entre la déformation désirée et celle finalement obtenue et le temps médian pour effectuer les calculs liés au suivi et à la loi de commande. Si le déplacement moyen est supérieur à l'erreur moyenne lorsque le câble est dans son état de repos, il est possible de savoir si la méthode sur-déforme le câble. L'erreur moyenne et l'erreur relative permettent d'évaluer la précision de la méthode, tandis que le temps médian permet d'évaluer sa capacité à fonctionner en quasi-temps réel. Différentes expériences ont été conduites pour évaluer les performances de la méthode en générant des déformations utilisant des câbles faits de différents matériaux et couleurs. Ces expériences ont montré que la méthode proposée était capable de générer des déformations avec une précision similaire dans ces différentes situations, montrant de ce fait son indépendance vis à vis des propriété physiques de l'objet. Plusieurs expériences ont été menées afin de déterminer si la méthode pouvait permettre de générer de grandes déformations. Ces expériences ont montré qu'en définissant des tâches intermédiaires, la méthode pouvait être utilisée pour générer des formes 3D complexes.

A.4 Suivi et estimation de paramètres élastiques d'objets déformables

Contexte introductif

Simultanément suivre et manipuler des objets déformables restent à ce jour des défis scientifiques à relever. Pourtant, différentes applications pourraient grandement en bénéficier, telles que des robots humanoïdes interagissant avec des objets déformables, les tests de conformité dans la fabrication de produits ou pour de la palpation dans les soins de santé préventifs.

Le suivi d'objets déformables peut se faire en utilisant des méthodes où les déformations sont inférées par des modèles physiques ou par des modèles géométriques. Des méthodes basées sur des modèles géométriques telles que [NFS15] ou [Inn+16] proposent des mécanismes efficaces pour suivre des formes 3D complexes sans avoir besoin de données d'entraînement, mais ne permettent pas de simultanément suivre et estimer des paramètres physiques d'objets déformables. Parmi les méthodes de l'état de l'art basées modèles physiques, il est possible de citer les travaux de Zhang *et al.* dont la méthode de suivi de déformations permet aussi d'estimer les forces externes appliquées sur un robot déformable à l'aide d'une caméra de profondeur [Zha+19]. Une autre approche, proposée par Petit *et al.*, permet quant à elle de suivre les déformations d'un objet déformable quelconque dont on possède le maillage et les paramètres physiques [Pet+18]. Enfin, une méthode permettant de simultanément suivre les déformations d'un objet déformable et estimer ses paramètres d'élasticité a été proposée par Frank *et al.* [Fra+10]. Cependant, cette méthode fonctionne en boucle ouverte et elle ne permet pas d'estimer les forces de contact déformant un objet déformable à partir d'informations visuelles.

La manipulation d'objets déformables est facilitée si les paramètres physiques, tels que les paramètres d'élasticité, de l'objet sont connus. A ce jour, plusieurs méthodes d'estimation ont été proposées. On peut notamment citer les travaux de Sedef *et al.* qui propose une méthode utilisant un dispositif haptique et un capteur de force pour estimer les paramètres de viscoélasticité d'un tissu homogène [SSB06]. Un objet viscoélastique possède une réponse élastique instantanée, qui peut être modélisée par la loi de Hooke, en réponse à une force externe et une réponse visqueuse sur le long terme. Une méthode capable d'estimer les paramètres d'élasticité d'un objet non-homogène a été proposée par Bickel *et al.* [Bic+09]. Finalement, des méthodes basées apprentissage telles que [Wan+15] ou [Fon09] capables d'estimer les paramètres d'élasticité d'objets non-homogènes ont été proposées. Cependant, l'acquisition des données nécessaires à ce genre de méthodes nécessite beaucoup de temps.

Nos contributions au suivi et à la manipulation simultanés d'objets déformables sont les suivantes:

- une méthode de suivi des déformations qui ne nécessite pas de marqueurs d'aucune sorte;
- une méthode itérative d'estimation des paramètres d'élasticité;
- une méthode d'estimation des forces de contact exercées sur un objet déformable utilisant seulement des informations visuelles.

Notre approche, développée conjointement avec A. Sengupta et publiée dans [Sen+20], fonctionne en boucle fermée ce qui permet d'accomplir deux objectifs : simultanément suivre

les déformations d'un objet déformable et estimer certains de ses paramètres d'élasticité; ainsi que d'estimer des forces de contact exercées sur un objet déformable en utilisant seulement des informations visuelles.

Méthodologie

Pour permettre de simultanément suivre les déformations d'un objet déformable et estimer certains de ses paramètres d'élasticité, notre approche nécessite un maillage 3D grossier de l'objet ainsi que des mesures des forces de contact s'exerçant sur l'objet pendant la procédure d'estimation. La méthode de suivi de déformation met en correspondance un modèle de l'objet déformé dans une simulation basée physique avec un nuage de points obtenu à l'aide d'une caméra de profondeur. Pour ce faire, une estimation de la pose de l'objet par rapport à la caméra est estimée dans un premier temps pour mettre en correspondance le repère monde de la simulation avec le repère caméra. Dans un second temps, le suivi de déformations est achevé en déterminant les forces extérieures à appliquer à certains points du maillage de l'objet pour que le modèle simulé corresponde au nuage de points observé. La détermination de ces forces se fait en minimisant simultanément une distance point-à-plan et une distance dans l'image entre des points caractéristiques.

L'estimation des paramètres d'élasticité se fait de façon itérative à l'aide d'une méthode de minimisation utilisant une simulation basée physique. Entre deux itérations de la minimisation, les paramètres physiques de l'objet simulé sont variables tandis que les forces extérieures sont constantes et égales à celles mesurées par le capteur de force de l'effecteur robotique. La fonction de coût qui est minimisée est la somme des distances euclidiennes entre le résultat du suivi de déformations et l'état final de la simulation. Une fois que l'estimation a convergé, la méthode de suivi de déformations met à jour les paramètres d'élasticité qu'elle utilise dans sa simulation physique. Ce système en boucle fermée est illustrée dans la Figure A.4.

Lorsque l'estimation des paramètres d'élasticité converge vers les mêmes valeurs obtenues pour plusieurs expériences successives , il est possible d'utiliser la méthode de suivi des déformations de sorte à estimer les forces de contact s'exerçant sur l'objet à partir d'informations visuelles uniquement, comme illustré dans la Figure A.5. Pour ce faire, les paramètres d'élasticité estimés sont utilisés pour modéliser l'objet dans la simulation basée physique utilisée pour le suivi. On calcule alors la résultante des forces des points du maillage situés autour de la zone de contact.

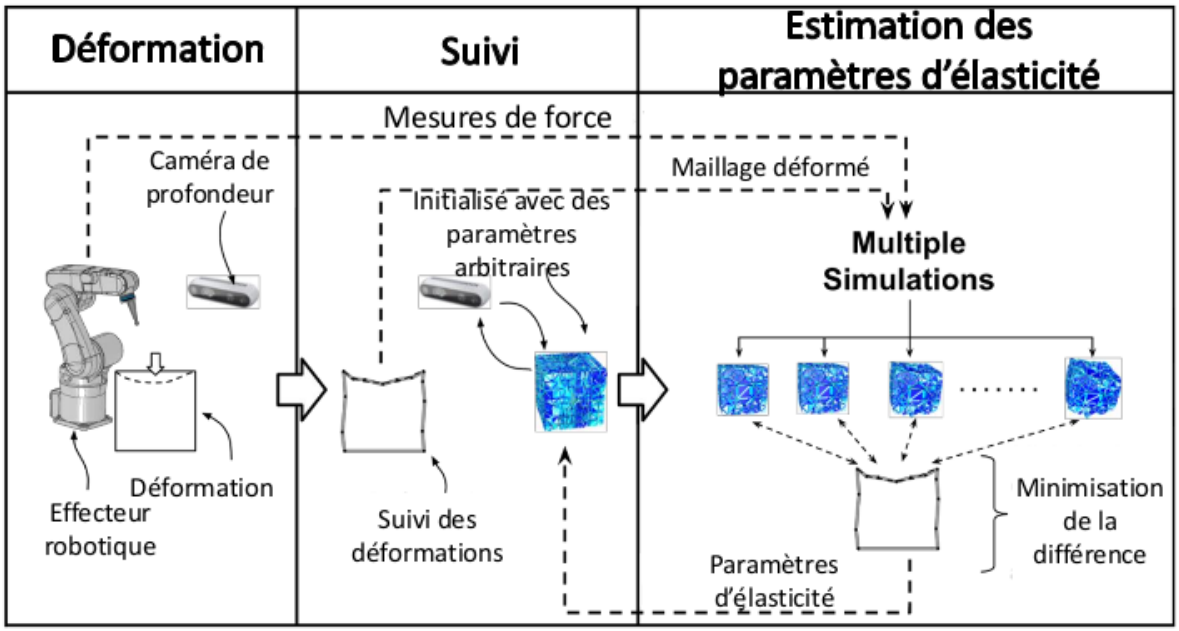


Figure A.4: Boucle fermée entre le suivi de déformations et l'estimation des paramètres d'élasticité.

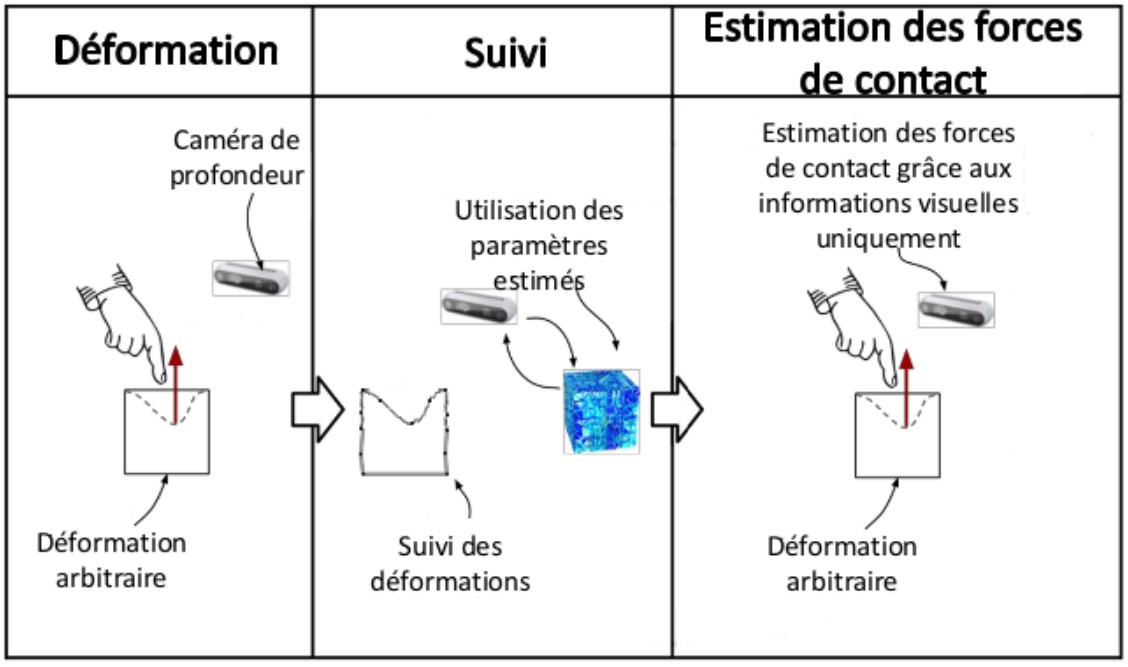


Figure A.5: Estimation des forces de contact appliquées sur l'objet en utilisant uniquement des informations visuelles. Pour cela, les paramètres d'élasticité estimés sont utilisés par la méthode de suivi des déformations.

Évaluation

Lors de l'évaluation de la méthode, seul le module de Young est estimé par la méthode d'estimation des paramètres d'élasticité. La précision de l'estimation du module de Young est évaluée dans différents cas : en faisant varier la résolution du maillage de l'objet testé et en faisant varier l'erreur entre le module de Young initial et le véritable. Des objets de formes et matières différentes ont été utilisés pour chacune de ces expériences. Le véritable module de Young de chacun de ces objets a été déterminé par une méthode classique de tests d'indentation. Les expériences visant à évaluer l'influence de la résolution du maillage sur l'estimation ont montré que plus le maillage était détaillé, plus l'estimation était précise. Les expériences faisant varier l'erreur entre le module de Young initial et le véritable ont montré qu'il était possible d'utiliser la méthode proposée même lorsque l'erreur initiale est importante. Toutes ces expériences ont montré que la méthode pouvait être utilisée sur des matériaux présentant des modules de Young de différents ordres de grandeur.

Pour finir, des expériences ont été menées pour évaluer la précision de l'estimation des forces de contact à partir d'informations visuelles uniquement. Ces expériences ont validé le fait que la méthode pouvait estimer des forces de contact à partir d'informations visuelles uniquement, ce qui pourrait s'avérer utile par exemple dans des situations de travail collaboratif entre un système automatisé et un opérateur humain.

A.5 Conclusion

Nous avons présenté le sujet de la manipulation robotisée d'objets déformables et nos contributions à relever certains des défis scientifiques qui y sont associés. Le premier défi scientifique abordé en détails dans ce manuscrit est la prédiction du comportement de déformations d'objets déformables. Le second défi scientifique abordé dans ce manuscrit découlant de la manipulation robotisée d'objets déformables est le contrôle précis de leur forme. Le dernier défi évoqué dans ce manuscrit est l'établissement de procédure permettant l'estimation de paramètres physiques d'objets déformables tout en assurant leur intégrité.

Dans un premier temps, nous avons présenté une méthode baptisée ADVISEd, qui est une méthode non-basée modèle de déformation active. La méthode utilise une matrice Jacobienne de déformation estimée en ligne afin de calculer la loi de commande permettant de déformer l'objet déformable. Cette matrice Jacobienne de déformation relie les déplacements de l'effecteur robotique aux déformations de l'objet. L'estimation est effectuée en utilisant une méthode de minimisation des moindres carrés avec fenêtre glissante. Afin d'améliorer la

robustesse de l'estimation face aux bruits de mesure, nous proposons d'ajouter un critère de confiance lors de la mise à jour de la matrice Jacobienne. Deux évaluations expérimentales de la méthode ont été effectuées. La première consiste à déformer un objet déformable en utilisant des marqueurs visuels pour effectuer le suivi de déformations. La deuxième consiste à effectuer des tâches de déformation ou de préservation de forme au repos d'objets déformables en utilisant une méthode de suivi de déformations ne reposant pas sur des marqueurs. Les différentes expériences ont montré que la méthode ADVISEd permet de contrôler les déformations d'un objet déformable à une vitesse quasi-temps réel tout en assurant une meilleure robustesse aux perturbations extérieures que des méthodes issues de l'état de l'art. Les expériences ont aussi montré que la méthode pouvait être utilisée pour générer de grandes déformations en divisant la tâche en tâches intermédiaires.

Dans un second temps, nous avons présenté une méthode permettant la manipulation précise de câbles déformables en utilisant deux bras manipulateurs robotiques. Afin de permettre la déformation active de câbles, nous proposons dans un premier temps une nouvelle caractéristique visuelle utilisant un modèle géométrique afin d'inférer la forme 3D d'un câble. Additionnellement, nous proposons une méthode de suivi de déformations ne nécessitant ni marqueurs ni modèles physiques. Cette méthode utilise du filtrage particulaire afin de suivre le câble à une vitesse quasi-temps réel. La caractéristique visuelle découlant du résultat du suivi est utilisée par notre méthode de déformation active pour calculer la loi de commande des deux robots. Cette méthode de déformation active est non-basée modèle et estime en ligne la matrice Jacobienne de déformation. Des expériences ont été menées sur des câbles dont les couleurs et matières diffèrent. La méthode proposée a permis de déformer le câble avec une précision similaire au cours de ces différentes expériences. D'autres expériences ont montré qu'il était possible d'utiliser cette méthode afin de générer des formes 3D complexes en définissant des tâches intermédiaires.

Finalement, nous avons présenté une méthode permettant simultanément de suivre les déformations d'un objet déformable et d'estimer son module de Young. Pour ce faire, l'objet est palpé à l'aide d'un effecteur robotique équipé d'un capteur de force. Cette opération se fait tout en observant la scène à l'aide d'une caméra de profondeur. Quand l'objet atteint un état d'équilibre, les résultats du suivi de déformation ainsi que les mesures de forces exercées sont utilisés afin d'estimer le module de Young. L'estimation se fait dans un processus de minimisation itératif utilisant des simulations basées physique. Les résultats de l'estimation sont ensuite réinjectés dans la méthode de suivi de déformation, formant ainsi un système en boucle fermée. Quand l'estimation du module de Young converge lors de plusieurs expériences vers la

même valeur, la méthode de suivi de déformation peut être utilisée pour estimer les forces de contact appliquées à l'objet en utilisant des informations visuelles seulement. Les expériences ont montré que la méthode pouvait être utilisée pour suivre et estimer le module de Young d'objets ayant des géométries et matières différentes. Ces expériences ont aussi montré que l'estimation des forces de contact pouvait être effectuée sur des objets complexes.

BIBLIOGRAPHY

[Ada+16] 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*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 2717–2722.

[Ala+18] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "A robust data-driven approach for online learning and manipulation of unmodeled 3-D heterogeneous compliant objects", *in*: *IEEE Robotics and Automation Letters* 3.*4* (2018), pp. 4140–4147.

[Alo+15] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 5495–5502.

[ASE02] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: Measuring errors between surfaces using the hausdorff distance", *in*: *Proc. of the IEEE Int. Conf. on multimedia and expo*, vol. 1, 2002, pp. 705–708.

[AW00] F. Abegg and H. Worn, "Robust algorithms for recognizing shape changes of deformable linear objects in video image sequences", *in*: *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 1, 2000, pp. 335–338.

[Beh+19] T. Behr, T. P. Pusch, M. Siegfarth, D. Hüsener, T. Mörschel, and L. Karstensen, "Deep Reinforcement Learning for the Navigation of Neurovascular Catheters", *in*: *Current Directions in Biomedical Engineering* 5.*1* (2019), pp. 5–8.

[Ber13] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 4525–4532.

[Bic+09] B. Bickel, M. Bächer, M. A. Otaduy, W. Matusik, H. Pfister, and M. Gross, "Capture and modeling of non-linear heterogeneous soft tissue", *in*: *ACM Transactions on Graphics*, vol. 28, 2009, p. 89.

[Bic+10] B. Bickel, M. Bacher, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, and W. Matusik, "Design and Fabrication of Materials with Desired Deformation Behavior", *in*: *Proc. of the SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010, 63:1–63:10.

[BM11] V. Buljak and G. Maier, "Proper Orthogonal Decomposition and Radial Basis Functions in material characterization based on instrumented indentation", *in*: *Engineering Structures* 33 (2011), pp. 492–501.

[Bro00] C. G. Broyden, "On the discovery of the "good Broyden" method", *in*: *Mathematical programming* 87.*2* (2000), pp. 209–213.

[Bro65] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations", *in*: *Mathematics of computation* 19.*92* (1965), pp. 577–593.

[CB19] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 6443–6450.

[CH06] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches", *in*: *IEEE Robotics and Automation Magazine* 13 (2006), pp. 82–90.

[CH80] N. Cressie and D. M. Hawkins, "Robust estimation of the variogram: I", *in*: *Journal of the Int. Association for Mathematical Geology* 12.*2* (1980), pp. 115–125.

[CKN15] P. Chatelain, A. Krupa, and N. Navab, "3D ultrasound-guided robotic steering of a flexible needle via visual servoing", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 2250–2255.

[CM00] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 1, 2000, pp. 630–635.

[Coe+15] E. Coevoet, N. Reynaert, E. Lartigau, L. Schiappacasse, J. Dequidt, and C. Duriez, "Registration by interactive inverse simulation: application for adaptive radiotherapy", *in*: *Int. journal of computer assisted radiology and surgery* 10.*8* (2015), pp. 1193–1200.

[De 72] C. De Boor, "On calculating with B-splines", *in*: *Journal of Approximation theory* 6.*1* (1972), pp. 50–62.

[Din+12] F. Ding, J. Huang, Y. Wang, T. Fukuda, and T. Matsuno, "Adaptive sliding mode control for manipulating deformable linear object with input saturation", *in*: *Proc. of the IEEE Int. Conf. on Mechatronics and Automation*, 2012, pp. 1862–1867.

[DL99] M. De Mathelin and R. Lozano, "Robust adaptive identification of slowly time-varying parameters with bounded disturbances", *in*: *Automatica* 35.*7* (1999), pp. 1291–1305.

[DS11] J. Das and N. Sarkar, "Autonomous shape control of a deformable object by multiple manipulators", *in*: *Journal of Intelligent and Robotic Systems* 62.*1* (2011), pp. 3–27.

[Fan+19] G. Fang, X. Wang, K. Wang, K.-H. Lee, J. D. Ho, H.-C. Fu, D. K. C. Fu, and K.-W. Kwok, "Vision-based online learning kinematic control for soft robots using local gaussian process regression", *in*: *IEEE Robotics and Automation Letters* 4.*2* (2019), pp. 1194–1201.

[Fau+07] F. Faure, J. Allard, S. Cotin, P. Neumann, P.-J. Bensoussan, C. Duriez, H. Delingette, and L. Grisoni, "SOFA: A modular yet efficient simulation framework", *in*: *Surgetica Computer-Aided Medical Interventions: tools and applications*, 2007, pp. 101–108.

[Fau+12] 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*, vol. 11, 2012, pp. 283–321.

[FB81] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *in*: *Communications of the ACM* 24.*6* (1981), pp. 381–395.

[Flu09] A. Fluent, "12.0 Theory Guide", *in*: *Ansys Inc* 5 (2009).

[Fon09] P. Fong, "Sensing, Acquisition, and Interactive Playback of Data-based Models for Elastic Deformable Objects", *in*: *The Int. Journal of Robotics Research* 28.*5* (2009), pp. 630–655.

[Fra+10] B. Frank, R. Schmedding, C. Stachniss, M. Teschner, and W. Burgard, "Learning the elasticity parameters of deformable objects with a manipulation robot", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 1877–1883.

[Fra+11] B. Frank, C. Stachniss, N. Abdo, and W. Burgard, "Efficient motion planning for manipulation robots in environments with deformable objects", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 2180–2185.

[GC15] F. Goulette and Z.-W. Chen, "Fast computation of soft tissue deformations in real-time simulation with Hyper-Elastic Mass Links", *in*: *Computer Methods in Applied Mechanics and Engineering* 295 (2015), pp. 18–38.

[GH03] N. R. Gans and S. Hutchinson, "An asymptotically stable switched system visual controller for eye in hand robots", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, 2003, pp. 735–742.

[GMN12] T. Garbowski, G. Maier, and G. Novati, "On calibration of orthotropic elastic-plastic constitutive models for paper foils by biaxial tests and inverse analyses", *in*: *Structural and Multidisciplinary Optimization* 46 (2012), pp. 111–128.

[GSS93] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *in*: *Proc. of the IEE Conf. F (radar and signal processing)*, vol. 140, 2, 1993, pp. 107–113.

[Hao+13] N. Haouchine, J. Dequidt, I. Peterlik, E. Kerrien, M.-O. Berger, and S. Cotin, "Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery", *in*: *Proc. of the IEEE Int. Symposium on Mixed and Augmented Reality*, 2013, pp. 199–208.

[Hao+14] N. Haouchine, J. Dequidt, I. Peterlik, E. Kerrien, and M.-O. Berger, "Towards an Accurate Tracking of Liver Tumors for Augmented Reality in Robotic Assisted Surgery.", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2014.

[HR05] F. Hild and S. Roux, "Digital Image Correlation: From Displacement Measurement to Identification of Elastic Properties - A Review", *in*: *arXiv:physics/0511122* (2005).

[HSP18] Z. Hu, P. Sun, and J. Pan, "Three-Dimensional Deformable Object Manipulation Using Fast Online Gaussian Process Regression", *in*: *IEEE Robotics and Automation Letters* 3.*2* (2018), pp. 979–986.

[Hub81] P.-J. Huber, *Robust Statistics*, ed. by Wiley, 1981.

[HYK11] M. Higashimori, K. Yoshimoto, and M. Kaneko, "Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation* 11 (2011), pp. 5120–5126.

[Inn+16] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, "VolumeDeform: Real-time volumetric non-rigid reconstruction", *in*: *Proc. of European Conference on Computer Vision*, 2016, pp. 362–379.

[Int20] Intel, *Tuning depth cameras for best performance*, 2020, URL: `dev.intelrealsense.com/docs/tuning-depth-cameras-for-best-performance`.

[JP11] M. Jancosek and T. Pajdla, "Multi-view reconstruction preserving weakly-supported surfaces", *in*: *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2011.

[KH11] S. B. Kesner and R. D. Howe, "Force control of flexible catheter robots for beating heart surgery", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 1589–1594.

[Kim+12] J.-H. Kim, A. Bartoli, T. Collins, and R. Hartley, "Tracking By Detection For Interactive Image Augmentation In Laparoscopy", *in*: *Biomedical Image Registration* (2012), pp. 246–255.

[LDH17] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tethered robots", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 732–738.

[LKM20a] R. Lagneau, A. Krupa, and M. Marchal, "Active Deformation through Visual Servoing of Soft Objects", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2020, pp. 8978–8984.

[LKM20b] R. Lagneau, A. Krupa, and M. Marchal, "Automatic Shape Control of Deformable Wires based on Model-Free Visual Servoing", *in*: *IEEE Robotics and Automation Letters* 5.*4* (2020), pp. 5252–5259.

[Low99] D. G. Lowe, "Object recognition from local scale-invariant features", *in*: *Proc. of the IEEE int. conf. on computer vision*, vol. 2, 1999, pp. 1150–1157.

[Mai+14] G. Maier, V. Buljak, T. Garbowski, G. Cocchetti, and G. Novati, "Mechanical characterization of materials and diagnosis of structures by inverse analyses: some innovative procedures and applications", *in*: *International Journal of Computational Methods* 11 (2014), p. 1343002.

[McK+11] C. T. McKee, J. A. Last, P. Russell, and C. J. Murphy, "Indentation versus tensile measurements of Young's modulus for soft biological tissues", *in*: *Tissue Engineering Part B: Reviews* 17.*3* (2011), pp. 155–164.

[Mee+91] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: A review", *in*: *International journal of computer vision* 6.*1* (1991), pp. 59–70.

[MF06] T. Matsuno and T. Fukuda, "Manipulation of flexible rope using topological model based on sensor information", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 2638–2643.

[MG04] M. Müller and M. Gross, "Interactive Virtual Materials", *in*: *Proc. of Graphics interface*, 2004, pp. 239–246.

[MH17] A. Malti and C. Herzet, "Elastic shape-from-template with spatially sparse deforming forces", *in*: *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 3337–3345.

[Mil+07] K. Miller, G. Joldes, D. Lance, and A. Wittek, "Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation", *in*: *Int. Journal for Numerical Methods in Biomedical Engineering* 23.*2* (2007), pp. 121–134.

[MMM12] P. Moulon, P. Monasse, and R. Marlet, "Adaptive Structure from Motion with a Contrario Model Estimation", *in*: *Proc. of the Asian Computer Vision Conference*, 2012, pp. 257–270, DOI: 10.1007/978-3-642-37447-0_20.

[Mor78] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory", *in*: *Numerical analysis*, 1978, pp. 105–116.

[MR93] H. Michel and P. Rives, "Singularities in the determination of the situation of a robot effector from the perspective view of three points", *in*: *Rapport de Recherche, INRIA 1850* (1993).

[MSC05] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills", *in*: *Robotics and Automation Magazine* 12.*4* (2005), pp. 40–52.

[MSJ07] A. Massoud Farahmand, A. Shademan, and M. Jagersand, "Global visual-motor estimation for uncalibrated visual servoing", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 1969–1974.

[Mül+02] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations", *in*: *Proc. of the SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002, pp. 49–54.

[MUS16] E. Marchand, H. Uchiyama, and F. Spindler, "Pose Estimation for Augmented Reality: A Hands-On Survey", *in*: *Transactions on Visualization and Computer Graphics* 22.*12* (Dec. 2016), pp. 2633–2651.

[Nav+13] D. Navarro-Alarcón, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators", *in*: *IEEE Transactions on Robotics* 29.*6* (2013), pp. 1457–1468.

[Nav+14] D. Navarro-Alarcón, Y.-H. Liu, J. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments", *in*: *The International Journal of Robotics Research* 33.*11* (2014), pp. 1462–1480.

[Nav+16] 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", *in*: *IEEE Transactions on Robotics* 32.*2* (2016), pp. 429–441.

[Nea+06] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics", *in*: *Computer Graphics Forum* 25 (2006), pp. 809–836.

[NFS15] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time", *in*: *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 343–352.

[NL18] D. Navarro-Alarcón and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-D image contours", *in*: *IEEE Transactions on Robotics* 34.*1* (2018), pp. 272–279.

[PB12] D. Pizarro and A. Bartoli, "Feature-based deformable surface detection with self-occlusion reasoning", *in*: *International Journal of Computer Vision* 97 (2012), pp. 54–70.

[Pet+17a] A. Petit, F. Ficuciello, G. A. Fontanelli, L. Villani, and B. Siciliano, "Using physical modeling and RGB-D registration for contact force sensing on deformable objects", *in*: 2017.

[Pet+17b] A. Petit, V. Lippiello, G. A. Fontanelli, and B. Sicialiano, "Tracking elastic deformable objects with an RGB-D sensor for a pizza chef robot", *in*: *Robotics and Autonomous Systems* 88 (2017), pp. 187–201.

[Pet+18] A. Petit, S. Cotin, V. Lippiello, and B. Siciliano, "Capturing deformations of interacting non-rigid objects using rgb-d data", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 491–497.

[PH12] N. Padoy and G. D. Hager, "Deformable Tracking of Textured Curvilinear Objects", *in*: *Proc. of British Machine Vision Conference*, 2012, pp. 1–11.

[PK17] P. Patlan-Rosales and A. Krupa, "Strain estimation of moving tissue based on automatic motion compensation by ultrasound visual servoing", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2017.

[PKK93] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision", *in*: *IEEE transactions on robotics and automation* 9.*1* (1993), pp. 14–35.

[PLS15] A. Petit, V. Lippiello, and B. Siciliano, "Real-time tracking of 3D elastic objects with an RGB-D sensor", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 3914–3921.

[PM06] M. Pressigout and E. Marchand, "Real-time 3d model-based tracking: Combining edge and texture information", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 2726–2731.

[Pra+10] P. Pratt, D. Stoyanov, M. Visentini-Scarzanella, and G.-Z. Yang, "Dynamic guidance for robotic surgery using image-constrained biomechanical models", *in*: *Medical Image Computing and Computer-Assisted Intervention* (2010), pp. 77–85.

[Qi+14] P. Qi, C. Liu, L. Zhang, S. Wang, H.-K. Lam, and K. Althoefer, "Fuzzy logic control of a continuum manipulator for surgical applications", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, 2014, pp. 413–418.

[RC11] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011.

[Roy+17] 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.", *in*: *Medical Image Analysis* 35 (2017), pp. 582–598.

[San+18] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey", *in*: *The Int. Journal of Robotics Research* 37.7 (2018), pp. 688–716.

[San+20] J. Sanchez, K. Mohy El Dine, J. A. Corrales Ramon, B.-C. Bouzgarrou, and Y. Mezouar, "Blind Manipulation of Deformable Objects based on Force Sensing and Finite Element Modeling", *in*: *Frontiers in Robotics and AI* 7 (2020), p. 73.

[Sch+13] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds", *in*: *Proc of the IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 1130–1137.

[Sen+20] A. Sengupta, R. Lagneau, A. Krupa, E. Marchand, and M. Marchal, "Simultaneous Tracking and Elasticity Parameter Estimation of Deformable Objects", *in*: *Proc. of the Int. Conf. on Robotics and Automation*, 2020, pp. 10038–10044.

[SFJ10] A. Shademan, A.-M. Farahmand, and M. Jagersand, "Robust jacobian estimation for uncalibrated visual servoing", *in*: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 5564–5569.

[SH02] A. Schlechter and D. Henrich, "Manipulating deformable linear objects: Manipulation skill for active damping of oscillations", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 2, 2002, pp. 1541–1546.

[SJ05] M. Shahamiri and M. Jagersand, "Uncalibrated visual servoing using a biased newton method for on-line singularity detection and avoidance", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 3953–3958.

[SKM19] A. Sengupta, A. Krupa, and E. Marchand, "Tracking of Non-Rigid Objects using RGB-D Camera", *in*: *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics*, 2019, pp. 3310–3317.

[Sla+17] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic, "Killingfusion: Non-rigid 3d reconstruction without correspondences", *in*: *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 1386–1395.

[SP09] J. Smolen and A. Patriciu, "Deformation planning for robotic soft tissue manipulation", *in*: *Proc. of the IEEE Int. Conf. on Advances in Computer-Human Interactions*, 2009, pp. 199–204.

[SSB06] M. Sedef, E. Samur, and C. Basdogan, "Real-time finite-element simulation of linear viscoelastic tissue behavior based on experimental data", *in*: *IEEE Computer Graphics and Applications* 26.*6* (2006).

[Thu97] O. Thual, *Introduction à la mécanique des milieux continus déformables*, Cépaduès, 1997.

[TT18] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration", *in*: *International Journal of Robotics Research* (2018).

[TWT18] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift", *in*: *IEEE Robotics and Automation Letters* 3.*4* (2018), pp. 3426–3433.

[Ver+19] M. Verghese, F. Richter, A. Gunn, P. Weissbrod, and M. Yip, "Model-free Visual Control for Continuum Robot Manipulators via Orientation Adaptation", *in*: *arXiv preprint arXiv:1909.00450* (2019).

[Wan+15] B. Wang, L. Wu, K. Yin, U. Ascher, L. Liu, and H. Huang, "Deformation Capture and Modeling of Soft Objects", *in*: *ACM Transactions on Graphics* 34.*4* (2015), 94:1–94:12.

[WH11] Z. Wang and S. Hirai, "Modeling and estimation of rheological properties of food products for manufacturing simulations", *in*: *Journal of food engineering* 102.*2* (2011), pp. 136–144.

[Zai+17] L. Zaidi, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, and L. Sabourin, "Model-based strategy for grasping 3D deformable objects using a multi-fingered robotic hand", *in*: *Robotics and Autonomous Systems* 95 (2017), pp. 196–206.

[Zha+17] Z. Zhang, T. M. Bieze, J. Dequidt, A. Kruszewski, and C. Duriez, "Visual Servoing Control of Soft Robots based on Finite Element Model", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, 2017.

[Zha+19] Z. Zhang, A. Petit, J. Dequidt, and C. Duriez, "Calibration and External Force Sensing for Soft Robots using an RGB-D Camera", *in*: *IEEE Robotics and Automation Letters* 4.*3* (2019), pp. 2356–2363.

[Zhu+18] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables", *in*: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 479–484.

[Zhu+19] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic Manipulation Planning for Shaping Deformable Linear Objects With Environmental Contacts", *in*: *IEEE Robotics and Automation Letters* 5.*1* (2019), pp. 16–23.

GLOSSARY

ANN Artificial Neural Network. 64

CPD Coherent Point Drift. 35, 36, 41

CT Computed Tomography. 67, 68

DIC Digital Image Correlation. 64

DOF Degrees Of Freedom. 20, 23, 24, 32, 53, 64, 73, 74, 79, 80, 82, 97, 105, 106, 108, 178

EM Expectation Maximization. 65

FEM Finite Element Method. 29, 31, 33, 38, 39, 46–48, 58, 60–65, 67, 120, 175

IBVS Image-Based Visual Servoing. 19–21, 23, 25

ICP Iterative Closest Point. 40, 41, 57

LLS Local Least-Squares. 25

MAP Maximum A Posteriori. 65

MLS Moving Least-Squares. 37

PBVS Position-Based Visual Servoing. 19, 21, 22, 25, 47, 175

PDE Partial Differential Equation. 29, 31

POD Proper Orthogonal Decomposition. 60, 63, 64

RBF Radial Basis Functions. 60, 62, 63

RGB-D RGB-Depth. 34, 38, 57, 61, 77, 80, 89, 98, 99, 105, 106, 108, 119, 121, 134–136

rheological having both elastic properties and pplastic properties.. 59

ROI Region Of Interest. 61, 62

SLAM Simultaneous Localisation And Mapping. 34

SoI Surface of Interest. 77, 84, 85, 87, 90

TLED Total Lagrangian Explicit Dynamic. 39

TRA Trust Region Algorithm. 60, 64

UIBVS Uncalibrated Image-Based Visual Servoing. 23–25

MATHEMATICAL SYMBOLS

---

a set of additional knowledge about the system, such as the camera intrisics parameters. 18, 20, 21

ω angular velocity vector. 9, 19, 20, 22, 100, 104

e_{Av} Average error, which corresponds to the average point-to-point euclidean distance between the desired position of a point and its current position.. 107, 108, 111, 113

c Control point of a B-spline. 99–101

D damping matrix (if followed by a subscript, it describes an element of the mesh of the object, otherwise it describes the whole object). 30, 31

x position after displacement and/or deformation of a point belonging to the object. 26, 32, 33, 124

u displacement vector field. 26, 29–32, 175

q state vector containing all the degrees of freedom. 23, 24

$\mathbf{v}^{\mathbf{e}}$ effector velocity vector with regard to the camera Cartesian frame, which groups the linear and the angular velocities. 73–76, 104

e error vector. 18–22, 73, 77–79, 81, 84, 85

$\mathbf{e}^{\mathrm{E}}$ Estimation error vector. 124

f force vector the object is subject to. If the object is rigid, these forces are external only, otherwise it comprises also the internal forces. 27–32, 121, 124, 175

Υ forgetting factor in a least-squares minimization method with sliding window. 74, 75, 83, 85, 86, 106

$\mathcal{F}_c$ current camera Cartesian frame. 21, 22

$\mathcal{F}_c^*$ desired camera Cartesian frame. 21, 22

$\mathcal{F}_0$ reference Cartesian frame. 21

$\mathbf{G}$ Matrix that contains the Gaussian weights that are applied to the neighboring points of the model current evaluated point. 103, 104

Ω Size of the window used to compute the Gaussian weighted significance of a point. 102–104

g Gaussian weighted distance between a model point and its closest point in the point cloud. 103, 104

$\mathbf{I}$ Identity matrix. 27

$\mathbf{im}$ image measurements vector, such as coordinates. 18, 20

$\mathbf{J}$ Jacobian matrix containing the first-order partial derivatives of a vector-valued function. 19–23

$\hat{\mathbf{J}}$ Jacobian matrix containing the first-order partial derivatives of a vector-valued function. 20, 23, 24, 73–79, 84, 104, 105

$\boldsymbol{v}$ linear velocity vector. 9, 19, 20, 22, 100, 104

$\mathbf{M}$ mass matrix of the object. 29, 30

$\mathbf{m}$ coordinates of a point belonging to the object, expressed in the object rest shape Cartesian frame. 26

$\overline{\mathbf{C}}$ material tensor of order four that relates the strain to the stress. 27

e_{ms} Mean square error. 85, 86

$\boldsymbol{\nu}$ noise on measurement vector. 74, 101

m Number of DOFs of the effector. 73–80, 84, 104

η Number of feature points. 20, 76–78, 80, 85, 99, 100, 104, 106, 107, 111

χ Number of particles used in the SIR filter. 100, 104, 106

$\mathbf{p}$ Real point belonging to the segmented point cloud.. 100, 102–104

$\hat{\psi}$ Optimal B-spline resulting from the SIR filter that models a wire. 104

u horizontal pixel coordinate. 20

v vertical pixel coordinate. 20

ν Poisson's ratio. 27, 58

$\mathbf{x}$ position vector (if followed by a subscript, it describes an element of the mesh of the object, otherwise it describes the whole object). 9, 19, 79

$\overline{e_n}$ Relative error, which corresponds to $\frac{100}{n}\sum_{i=1}^{n}\frac{\|m_i^*-m_i(0)\|-\|m_i(tf)-m_i(0)\|}{\|m_i^*-m_i(0)\|}$.. 107–109, 111

$\mathbf{x}_0$ rest shape of the object, expressed in the world Cartesian frame. 26, 32, 33

$\mathbf{C}$ Damping matrix to use an energy-motivated dynamic-state feedback control. 79, 80, 83, 85

$\mathbf{R}_e$ rotation matrix of an element of a mesh used in Corotational FEM. 32, 33

σ_w Standard deviation of the Gaussian function used to apply weights the neighbors of an evaluated model point. 103, 104

σ_g Standard deviation of the Gaussian function used in the Gaussian weighted distance function. 103

W Significance of the whole model with regard to the segmented point cloud. 104

w Significance of a model point with regard to the segmented point cloud. 102, 104

$\hat{\mathbf{p}}$ Result of the SIR tracker, i.e. weighted sum of the particles.. 104, 107

n Size of the deformation feature vector. 18, 73, 74, 77–79, 83

θ The order of a spline. 99, 101, 106

$\mathbf{p}$ Numerical state vector providing a smooth trajectory to the effector. 79, 80, 84

$\boldsymbol{\xi}$ Stiffness parameters of an object. 123, 124

$\mathbf{K}$ stiffness matrix (if followed by a subscript, it describes an element of the mesh of the object, otherwise it describes the whole object). 30–32

$\overline{\boldsymbol{\epsilon}}$ strain tensor of order two. 26, 27, 29, 30

$\overline{\boldsymbol{\sigma}}$ stress tensor of order two. 27, 29, 30

${}^{c^*}\boldsymbol{t}_c$ coordinates of the origin of the current camera Cartesian frame expressed relatively to the desired camera Cartesian frame. 22

${}^{c}\boldsymbol{t}_0$ coordinates of the origin of the object Cartesian frame expressed relatively to the current camera Cartesian frame. 21, 22

${}^{c^*}\boldsymbol{t}_0$ coordinates of the origin of the object Cartesian frame expressed relatively to the desired camera Cartesian frame. 21, 22

α velocity control law gain. 77–80, 83–86, 104, 106

$\mathbf{v}^{\mathbf{ctrl}}$ velocity law vector used for eye-to-hand visual servoing. 19, 20, 22, 73, 77–79, 84, 104

$\mathbf{v}$ velocity vector the object, i.e the derivative of its position with regard to the time.. 9, 19

$\mathbf{s}$ visual feature vector. 18–24, 73–75, 77–79, 84, 99, 104

N size of the window in a least-squares minimization method with sliding window. 74, 75, 83, 85, 86, 106

ψ B-spline that models a wire. 99, 100, 104

$\tilde{\mathbf{p}}$ Point that belongs to a B-spline that models a wire. 99, 100, 102–104, 107

E Young modulus. 27, 28, 58, 124, 127, 129, 130

List of Figures

List of Tables

Titre : Contrôle de la forme d'objets déformables par asservissement visuel adaptatif

Mots clés : Objets déformables, contrôle de forme, asservissement visuel adaptatif, estimation de paramètres de rigidité

A ce jour, la manipulation robotisée d'objets non-rigides est bien plus complexe que celle d'objets rigides du fait des déformations que subissent ces objets lorsque des forces extérieures leur sont appliquées. Améliorer la manipulation robotisée de ce type d'objets est cruciale pour des applications telles que la production de denrées alimentaires ou en chirurgie. Parmi les défis scientifiques importants résultant de la manipulation robotisée d'objets non-rigides, nous avons choisi de concentrer nos travaux sur le contrôle de leur forme et l'estimation de leurs propriétés physiques.

Dans un premier temps, nous proposons une nouvelle méthode de contrôle de déformation qui ne nécessite pas de connaissances concernant les propriétés physiques de l'objet. Cette méthode estime en ligne la matrice Jacobienne de déformations qui relie les déplacements du ou des effecteur(s) robotique(s) aux déformations résultantes. En utilisant cette matrice, il est possible de calculer la loi de commande permettant de générer la déformation désirée. Pour évaluer la précision et la stabilité de la méthode, nous proposons une évaluation reposant sur des marqueurs visuels et une ne reposant pas sur des marqueurs.

Dans un second temps, nous présentons une méthode permettant une manipulation de câbles déformables à l'aide de deux bras manipulateurs robotiques. Pour ce faire, nous proposons une nouvelle caractéristique visuelle basée sur un modèle géométrique permettant de décrire la forme 3D d'un câble déformable, ainsi qu'une méthode de suivi permettant de suivre la forme 3D de câbles déformables en temps quasi-réel. Nous proposons également une méthode de contrôle de déformations non-basée modèle pour commander les manipulateurs robotiques tenant les extrémités du câble.

Enfin, nous proposons une méthode permettant simultanément de suivre les déformations d'un objet déformable et d'en estimer la rigidité. En utilisant les mesures de forces de contact et le résultat du suivi des déformations, la rigidité est estimée via une procédure d'optimisation. Les propriétés physiques utilisées pour le suivi de déformations sont mises à jour en utilisant le résultat de l'optimisation, formant ainsi un système en boucle fermée. Quand l'estimation de la rigidité converge pour plusieurs expériences successives, la méthode de suivi de déformation peut être utilisée afin d'estimer les forces de contact uniquement à partir d'informations visuelle.

En améliorant la manipulation robotisée d'objets déformables, ces travaux contribueront au développement de robots capables d'adapter automatiquement leur comportement face à des objets inconnus, leur permettant de remplacer des opérateurs humains dans des environnements inhospitaliers.

Title : Shape control of deformable objects by adaptive visual servoing

Keywords : Shape control, adaptive visual servoing, robotic manipulation, parameter estimation

Abstract: Nowadays, robots are able to manipulate rigid objects at a high rate and with an accuracy greater than human operators. However, when it comes to non-rigid objects, dexterous robotic manipulations become much difficult due to the deformations that occur when forces are applied. Improving the manipulation of such objects is crucial for applications such as food manufacturing or surgery. Among the scientific challenges arising from the robotic manipulation of deformable objects, we chose to focus our work on controlling their shape and estimating their physical properties.

First, we propose a novel deformation servoing method allowing to control the shape of an object while not requiring any knowledge on its physical properties. We estimate online the deformation Jacobian matrix that relates the motions of the end-effector(s) to the resulting deformations. Using this matrix, it is possible to compute the velocity control law that permits to generate a desired deformation. To evaluate the accuracy and stability of the method, we also propose both a marker-based and a marker-less evaluations.

Then, we present a method permitting dual-arm robotic manipulation of deformable wires. To this end, we propose a novel visual feature based on a geometric model to infer the shape of a deformable wire. Along with this novel visual feature, we propose a wire tracking method that permits to infer the 3D shape of a deformable wire at an interactive time. Finally, a model-free deformation servoing is used to control two end-effectors handling the extremities of the wire.

Finally, we propose a method to simultaneously track a non-rigid object and estimate its stiffness. Using the deformation tracking results and force measurements, the stiffness parameter is estimated in an optimization procedure. The physical parameters used for deformation tracking are updated using the estimated value, closing the loop. When convergence is reached, the deformation tracking can be used as a remote force sensor.

By improving robotic manipulation of deformable objects, these contributions pave the way for the development of robots able to automatically adapt their behavior with unknown deformable environments, allowing them to replace human operators in harmful environments.