IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2023, London, UK. Cite as T-RO paper.

Interaction Control of a Robotic Manipulator with the Surface of Deformable Object

Athanasios C. Dometios, Costas S. Tzafestas
School of Electrical and Computer Engineering, National Technical University of Athens, Greece
athdom@mail.ntua.gr, ktzaf@cs.ntua.gr

***Abstract*—Robotic manipulation of deformable objects has drawn the attention of researchers over the past few years and is associated to a large spectrum of new application perspectives. In this paper, we present an efficient integrated motion planning framework to effectively and accurately control a robotic manipulator executing interactive tasks on the surface of a deformable object. The proposed interactive motion planning framework is based on a mesh representation of the object, integrating three efficient preprocessing algorithmic steps, including visual object segmentation, FEM deformation tracking and local mesh parameterization. The use of barycentric coordinates, defined on the mesh triangles, enables the establishment of bijective transformations between the deformable part of an object surface and its planar (static and dynamic) parameterized mapping. By merging these spatial transformations with the preprocessing steps, in combination with an active stiffness scheme for robot manipulator control, we are able to achieve accurate and reactive motion planning of interactive trajectories, even under large and persistent visual occlusions (such as due to the presence of the robot in the visual scene). An extensive experimental evaluation study is presented, involving a robotic manipulator in interaction with a hemispherical model of controllable periodic active deformation, which permits precise ground truth derivation. Motion planning accuracy is evaluated in comparison with our previous direct vision-based approach, showing clearly superior performance of the proposed approach under all experimental conditions. The performance of the proposed framework is also further highlighted in tasks involving physical point tracking, interactive programming by human demonstration, as well as contact force regulation.**

I. Introduction

Autonomous interaction of robotic manipulation systems with various objects constitutes undoubtedly a vast research topic, which has attracted a lot of attention and research effort for nearly three decades now. Despite the fact that all these efforts have made significant progress in the direction of rigid object manipulation, which is now considered as a mature field in robotics [1], the study of deformable objects still remains limited and constitutes a challenging research field, as also described in a recent review paper [2]. In real world applications, the type of materials that a robot is called to interact with are usually non-rigid, since our world is formed mainly of deformable materials, the flexibility of which varies significantly. In order to achieve accurate and efficient robotic manipulation, a complete modelling of the manipulated object is required and coupled with the control process on-line.

Early attempts to model the behaviour of deformable objects have been made in the field of computer graphics, where the goal is to produce more realistic animations. Nealen et al. in [3] and Moore et al. in [4] offer a comprehensive introduction to the computer graphics methodologies of the previous decade in their survey papers. A variety of techniques has been applied in these early approaches to achieve deformation modeling. In most cases, an object's shape representation (e.g. a set of particles or a mesh) and a deformation model synthesize a deformation estimation technique. A mesh representation consists of a set of points (vertices), edges and faces, which are usually triangles or quadrilaterals, or elements, the representation of which is commonly made by tetrahedra or hexahedra. The deformation models are essentially functions, which calculate the position of every vertex taking into account both their current state (position and velocity) and an input force. Different deformation models have also been developed which do not require a mesh but work directly on particles either in 2D [5] or in 3D [6].

The categorization of the mesh-based models is formed either as continuous or as discrete variable models. The main representative of discrete methods is Mass-Spring-Damper (MSD) systems, in which the vertices of the mesh are treated as mass particles and the edges are considered as springs modeling the flexibility of the object. On the other hand, continuum models are usually set up with finite element methods (FEM), in which the object's shape is approximated by a set of discrete geometric parts called finite elements. Although FEM-based models produce more physically realistic deformation results, MSD models have gained a lot of researchers' attention, since they are more intuitive and simpler to implement. However, the processing power has increased significantly, making nowadays possible the real time simulation of realistic deformation (e.g. using FEM) of solid objects. This technological step forward has also shifted the interest of robotics researchers to the study of various interaction tasks such as manipulation, grasping and object identification. Recent survey papers [7], [8], [9] describe this research trend, with a significant amount of novel work now emerging in this field.

This paper capitalises on recent advances regarding real-time deformation modeling, to propose a novel perception-driven motion planning framework for effectively controlling a robotic manipulator when executing interactive tasks on the surface of a deformable object. The proposed framework combines 3D perception and on-line deformation modeling, with real-time motion planning and interaction control, enabling the system to efficiently handle both active and passive deformation scenarios. The core of the approach is based on

real-time FEM deformation tracking and efficient local mesh parameterization which, combined with stiffness control of the robotic manipulator, enables accurate reactive planning of interactive trajectories and contact force regulation on the surface of a dynamically deforming object, even in the presence of visual occlusions.

The paper is organized as follows. Section II describes related work and outlines the contribution of the paper. Section III formulates the problem while Section IV presents the proposed interactive planning framework along with the required mathematical background. Extensive experimental validation results are presented in Section V, involving a periodically deforming hemispherical model interacting with a robotic manipulator. Trajectory planning accuracy is evaluated in comparison with our previously proposed vision-based approach. Results regarding physical point tracking, interactive task efficacy and contact force regulation are presented for validation of the proposed approach. Finally, conclusive statements and future work directions are provided in Section VI.

II. Related Work

The first step of a robotic system interacting with its environment is the scene perception and understanding. Visual input is the most common perception method for robots, especially after the evolution RGB-D cameras, which provide rich optical information. Depth information is combined with camera motion in KinectFusion and DynamicFusion methods [10], [11] and more recently in InfiniTAM project [12] to generate a 3D mesh of the environment in real-time. In [13] Zollhöfer et al. registers RGB-D data on a rigid template, in order to capture the deformations of general shapes, while recently Güller and Kokkinos demonstrated in [14] the power of Deep Learning in accurate human body shape reconstruction even with a monocular camera. Another template-based framework proposed in [15] is combined with visual features from the texture of the object to track the deformations of objects, which may undergo topological changes (e.g. paper tearing), whereas Famouri et al. [16] employ the same ideas for tracking deformations using monocular camera as an input. Additionally, Willimon et al. integrate classic computer vision techniques with energy function minimization in [17], to estimate the configuration of a non-rigid cloth-like object. NURBS functions from computer graphics are also used together with visual features for deformation tracking in [18].

Recent papers involve a robotic system into deformation tracking tasks. More specifically, in [19] the authors characterize generic deformable objects by visually observing their interaction with a robotic hand and in [20] surface variations caused by the contact of a simulated robotic hand are measured visually on Point Cloud data. Tian and Jia in [21] extend their work on shape modelling of shell-like objects, which are grasped by a robotic hand. Visual depth data together with force-torque obtained from a sensor mounted on a probe, are fed into a neural gas network impemented in [22] to predict the deformation's characteristics of an object, without requiring knowledge of the object material. Real-time deformation tracking, which is the focus of [23] and more recently of [24], [25] is a prerequisite for closed-loop robotic manipulation of objects such as ropes, clothes or sponges.

Although deformation tracking is crucial for the success of a robotic task, it is not enough for a robotic object manipulation scenario. Different control and motion planning strategies have been developed for different kinds of objects. A work of Tanner et al. in [26] studies the control aspects of two mobile manipulators carrying a deformable cloth-like object. In general, garments and cloth-like objects have drawn a lot of research attention, due to their easier modeling than other 3D deformable objects and clear visual structure. Yinxiao Li et al. in a series of papers propose solutions for various interaction tasks with garments such as real-time pose estimation [27], unfolding [28] and folding [29]. Moreover, a complete pipeline of the autonomous folding task is presented in [30], which includes a vision-based garment grasping from a pile of clothes by a dual arm robot, classification and proper unfolding on a table (previously presented in [31]) and a folding technique executed by the robotic system based on dynamic programming. Additionally, in [32] a geometric approach for garment unfolding is proposed, taking advantage of the clear visual structure of clothes to estimate appropriate grasping points. Another challenging task is the autonomous paper origami folding which was implemented in [33], employing a two-handed multi-fingered robot and depth vision system. A dual arm robot is also developed in [34] for industrial automatic electronic soldering tasks on flexible PCBs.

Despite the large amount of work focusing on planar or even cloth-like objects, the robotic manipulation of solid deformable objects (e.g. sponges, plush toys and food) hasn't made the same progress, mainly due to the computational cost imposed by the simulation of 3D objects. However, the development of real-time deformation simulation techniques such as FEM and MSD has changed this picture. Examples of such approaches are [35], in which a FEM-based model is combined with a vision system in order to accomplish the object's shape modification by a robotic arm, and [36], in which the object is modelled as a mass-spring-damper system undergoing deformations by a robotic system in order to bring an internal point to a desired position. Object's internal point control is the main focus of [37], the authors of which propose a model-free controller with visual feedback for the estimation of the deformation Jacobian matrix. Another model-free method in [38] introduces the notion of diminishing rigidity in order to approximate the deformation Jacobian during manipulation tasks. A relevant problem concerns grasping of deformable objects, which also constitutes a particularly challenging robotic task, since the shape and consequently the grasping forces are modified during the grasping action. In order to address these problems, the authors of [39], [40] integrate the FEM modeling of the object with the tactile sensors of a multi-fingered robotic hand, to pick up objects from a table. Furthermore, visual and force data are combined with deformation modeling techniques in [41], [42], [43], to estimate (with the use of a robotic arm) the physical parameters of an object, which are very important for accurate modeling and interaction tasks. Caccamo et al. in [44] and Güler et al. in [45] use similar procedures with robotic probing on the surface of an object

but they implement visual meshless techniques to estimate the deformation of objects.

Soft tissue is another solid deformable object, which has been extensively studied for robotic medical applications. The constraints of these applications (e.g. real-time interaction with moving organs and limited action space during surgical tasks) are not met in household or industrial applications. Therefore, the robotic actions during medical procedures have to be more precise, delicate and well-planned. In order to fulfil the aforementioned requirements in medical scenarios, robotic systems need an accurate perception layer, which is frequently based on vision, as described in [46]. The main goal of these vision-based systems is to augment the visualization capabilities of the surgeon during minimally invasive surgery, with the use of pre-operative CT-scan data combined with a time-of-flight camera for the development of an organ's surface matching algorithm in [47], or a stereo endoscope and a biomechanical model, like the work presented in [48] for real-time motion tracking of the liver. The more challenging problem of monocular 3D reconstruction is faced in [49], in which a non-linear mechanical elastic model is proposed, to enable the reconstruction of highly deformable objects, presenting some promising results in computer assisted surgery scenarios, whereas in [50] and more recently in [51] the authors employ 3D template matching and Shape-from-Motion visual techniques for reconstruction of the peritoneal region and liver respectively.

Many surgical tasks require the interaction of a robotic manipulator with the surface of an organ or tissue. Papers involving robotic manipulators combine visual techniques with FEM physical simulation and a force sensor [52], in order to develop real-time tracking system of soft tissues' deformation produced by a manipulator. However, non-visual techniques proposed in [53] exploit the motion of a robot and more specifically motion planning algorithms and a hybrid force-motion controller, to estimate the shape and impedance parameters of tissue. Motion plans are also computed in [54] for a 6-DOF gripper for automated tissue retraction given the model of tissues in the vicinity of areas of interest. In a similar context concerning interaction with dynamically deforming organ tissues in surgical robotics, active motion compensation constitutes another open research topic, with a lot of effort focusing on beating heart surgery scenarios. In [55] a predictive force control algorithm is proposed to compensate priorly known motions generated both from breathing and beating heart, whereas a recent paper [56] proposes an impedance control scheme both for motion compensation in the slave robot and haptic feedback to the master robot in a tele-operation scenario. Another tele-operation approach, which integrates a visual system for motion perception, is presented in [57] using the da Vinci Research Kit (Intuitive Surgical Inc.). Its perception system estimates the distance from a rigid object (moved by an external device), which is used as reference for the da Vinci patient side manipulators to track motion in real time. Another example of visually guided surgical device is presented in [58]. Leonard et al. have developed STAR system, which includes a 7DOF robotic manipulator and a monocular camera and is dedicated to automate laparoscopic suturing. It uses vision guided system, through which the doctor can also determine the suturing points, and its accuracy and efficiency was tested on planar phantoms. Relevant interactive tasks are also met in industrial applications, with some recent research work starting to focus on how to handle similar constraints involving real-time physical interaction with the surface of dynamically deforming objects in a robot cell, like in [59] where a one dimensional FEM model is used for bimanual robotic cleaning of a deformable object.

Although a lot of research effort has been spent on both perception and control algorithms for smooth robotic manipulator interaction with deformable objects, no integrated framework exists to date, to the best of our knowledge, which combines: (a) enhanced visual perception, (b) reactive task planning and (c) contact force regulation, dealing with both active and passive deformation scenarios. In this paper, we propose an interactive mesh-based framework integrating 3D perception and on-line deformation modeling, with real-time motion planning and interaction control. The goal is to effectively and accurately control a robotic manipulator when executing interactive tasks on the surface of a deformable object. The main contributions of this work are:

- The transformation of the interaction control problem from the 3D Physical Space onto two distinct 2D canonical spaces: the Static Canonical Space (SCS) and the Dynamic Canonical Space (DCS), represented by two mesh structures which used to interactively plan the motions of the robot on a task space relative to the target surface area of a deforming object. The use of the DCS, in particular, and its associated continuously updated mesh structure enables geodesic distance preservation on the surface of a deforming object, which constitutes an important feature of the proposed interactive motion planning framework.
- The combination of continuous mesh parameterization with barycentric coordinates to navigate on the mesh triangles, also employing a normal shading technique to ensure continuity of motion parameters across the mesh elements. The use of barycentric coordinates and mesh parametrization, as defined in our approach, ensures bijectivity of the transformations between the physical and the canonical control spaces, enabling the generalization to more complex areas of deformable surfaces, which constitutes an important contribution with respect to prior work on relevant problems.
- The integration in the approach of an efficient FEM-based mesh tracking algorithm, combining prediction and perception-driven update steps, enabling the mesh structures to constantly track and optimally fit on the Point-Cloud data as perceived by a visual RGB-D sensor. This step is crucial for the performance of the proposed framework, both in terms of real-time efficiency and accuracy, but also because it enables to properly handle situations where large visual occlusions exist, where other systems actually fail.

The combination of the proposed motion planning approach with a geometrically consistent active stiffness control scheme further enables direct generalisation of our framework to contact tasks that also require force regulation during physical interaction with an actively deforming surface. The main challenges being addressed by the proposed framework concern

both: (a) the lack of prior knowledge regarding an explicit dynamic deformation model, which could readily assist the online tracking process, in combination (b) with the presence of large and persistent visual occlusions, such as those caused by the inevitable presence of the robot manipulator in the visual interaction scene. These are really hard challenges to tackle in real-time by existing methods. Our integrated framework constitutes one step towards handling such demanding tasks, in particular tasks requiring actual interaction of a robot manipulator with an actively deforming surface of a-priori unknown shape and mechanical properties. The performance of the proposed motion planning and control framework is demonstrated and validated on an extensive series of experiments, involving different interactive tasks performed on the surface of an homogeneously deforming spherical object.

III. Problem Formulation

The problem considered in this paper concerns the interaction control of a robotic manipulator executing a task on the surface of a curved and deformable object, in a workspace equipped with a depth-camera. The main objective of the manipulator is to navigate over a part of an object's surface and physically interact with it, while the object is actively or passively deforming. Although the initial shape of the object may be known, we assume that an explicit dynamical deformation model is not available. This assumption means that the evolution over time of the object's deformation (active or passive) is considered to be unmodeled, i.e. there is no function describing its motion over time; hence its state in the future cannot be predicted with precision and consequently the robot motion cannot be planned accordingly. The use of a depth camera and a physics-based tracking algorithm based on FEM, as performed in this work for object perception, is an initial step towards this direction. However, it still needs to be emphasized that we do not assume any knowledge of physically accurate elasticity parameters of the object, or any prior knowledge of the active dynamical deformation (periodic or not) performed by the object's surface. These assumptions along with the visual occlusion caused by the robot during the execution of interactive tasks evidently impose additional challenges to the problem at hand.

The main initial motivation for this work originates from an EU funded project, called i-Support [60], in which a soft-robotic manipulator was developed to perform interactive bathing tasks on the human body, actively compensating for the human body motion, either in contact (e.g. scrubbing) or non-contact (e.g. water pouring) interaction scenarios. However, besides these application scenarios, there exists a large variety of other applications relevant to the specific task setup considered in this work, which may also involve a robot executing a practical task over the surface or in contact with an object, either rigid or deformable. Focusing on the deformable scenarios, which are more challenging, one relevant application area could, for instance, focus on surgical tasks, such as those encountered during beating heart surgery, where robotic motion needs to be controlled over the surface of a continuously palpating and deforming surface. In such a task scenario, the goal is to actively compensate for the deformation of the target surface, with the use, in such a case, of an endoscopic camera as an input. In other relevant task scenarios, our approach could be used to actively compensate for the human respiratory motion during a robot assisted operation. Tasks involving manipulative actions on the surface of deformable objects and materials can also be found in industrial applications.

In all the above cases of interactive tasks, the control of the (physical or non-physical) interaction between the robot manipulator and the surface of the dynamically deforming object is crucial for the success of the planned task. In particular during contact, the object sets constraints on the paths that the end-effector can follow. Hence, if the task is not accurately planned, the use of a pure motion control strategy in constrained motion scenarios may lead to failure. Accurate planning would, successively, require an accurate model both of the robot and the object. Although, modelling of the manipulator can be achieved with high precision based on kinematic and dynamic analysis, the motion model of a deformable object is difficult to obtain even if the original shape is described in detail. The modelling errors will give rise to planning errors along time and in turn, contact forces will be applied on the end-effector causing a deviation from the desired trajectory. At the same time, the robot's control system will react to reduce such deviation, resulting in a build-up of the contact force, until saturation of the actuators is reached or breakage of the parts in contact eventuates.

The effect of the phenomenon described above can be decreased if a compliant behaviour is ensured during the interaction. Compliant interaction behaviour can be achieved either passively or actively. Passive compliance can be realized via the structural compliance of robot's links, joints (e.g. soft robot arms [61]) or end-effector (e.g. remote center of compliance for industrial applications [62]). Despite the fact that the passive approach is cheap and simple, it can only deal with small position and orientation deviations and cannot guarantee that high contact forces will never occur. On the other hand, active compliance can be established with either indirect force control schemes or direct force control schemes, which use as feedback force measurements describing the state of interaction. However, the lack of reliable force measurements in many real robotic systems has led us to the adoption of an indirect force control strategy. It should be pointed out here that the potential unavailability of force measurements concerns mainly the constraints related to the integration of such a device in specific robotic applications, such as some medical, surgical or even specific industrial ones. Of course, the additional use of force measurements combined with an impedance control scheme could be complementary and beneficial to the performance of the control framework proposed in this paper. The assumption of force/torque measurements unavailability made in this paper also enables us to highlight the capacity of our approach to regulate the contact forces with the use solely of visual perception combined with the proposed mesh structures, even with the absence of an f/t sensor, and even more in a visually occluded scene, as will be described later on in the paper. Therefore, in this work we have

adopted and used throughout the experiments a **geometrically consistent active stiffness** control scheme, described in [63], instead of any other indirect force control methods (such as an impedance control scheme). The approach followed in our work also takes into account the assumption regarding the unavailability of an explicit dynamical deformation model, as already explained at the beginning of this section.

A. Active Stiffness Control Preliminaries

Consider the case in which two rigid bodies, E representing the end-effector and D for the desired contact point on the surface, are elastically coupled. The reference frames Σ_e and Σ_d, which are attached to E and D respectively, coincide at the equilibrium. It is important to mention that the mechanical stiffness defined in this method, describes an ideal 6-DOF spring, which stores potential energy. An ideal stiffness has a potential energy function, which depends only on the relative pose of the two attached bodies and is port symmetric. The predominant behaviour of a physical 6-DOF spring is similar to an ideal one, however there are always parasitic effects causing energy dissipation.

In order to achieve a geometrically consistent 6-DOF active stiffness between the manipulator and the deformable object, a suitable control law is required. This control law should define the behaviour of the end-effector, under the application of an elastic wrench from the object. The expression and the properties of the elastic wrench for small displacements should be extended to the case of finite displacements. A finite displacement can be expressed, in the problem of robot - surface interaction control, as the relative pose of a desired frame Σ_d with respect to the end-effector frame. We can distinguish two cases: (a) The desired frame is put above the object's surface and hence the robot is not in contact with the object and (b) the desired frame is put below the object's surface, therefore the robot has contact with the object.

Introducing the robot kinematics and dynamics notation we start with the joint configuration vector and its time derivative: $\mathbf{q} := [q_1, ..., q_n]^T, \dot{\mathbf{q}} := [\dot{q}_1, ..., \dot{q}_n]^T \in \mathbb{R}^n$. The generalized velocity of the end-effector is denoted by the twist vector $\boldsymbol{\upsilon_e} = [\dot{\mathbf{p}}_\mathbf{e}^T \, \boldsymbol{\omega_e}^T]^T \in \mathbb{R}^6$ where $\dot{\mathbf{p}}_\mathbf{e}$ is the translational velocity and $\boldsymbol{\omega_e}$ the angular velocity. It can be computed through the differential kinematics equation [64]: $\boldsymbol{\upsilon_e} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$, where $\mathbf{J}(\mathbf{q}) : \mathbb{R}^n \rightarrow \mathbb{R}^{6\times n}$ is the robot's geometric Jacobian. We define also the set $\mathbb{D} := \{\mathbf{q} \in \mathbb{R}^n : \det(\mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T) > 0\}$ which contains all the singularity free configurations. The force $\mathbf{f_e}$ and the moment $\boldsymbol{\mu_e}$ applied by the end-effector to the environment are the components of the wrench $\mathbf{h_e} = [\mathbf{f_e}^T \boldsymbol{\mu_e}^T]^T$. In addition, the end-effector pose can be expressed as $\mathbf{x_e} = [\mathbf{p_e}^T \boldsymbol{\zeta_e}^T]^T \in \mathbb{R}^7$, where $\mathbf{p_e}$ denotes the position and $\boldsymbol{\zeta_e}$ denotes the orientation as a unit quaternion $\boldsymbol{\zeta_e} = [\phi_e \boldsymbol{\epsilon_e}^T] \in \mathbb{S}^3$ with $\phi \in \mathbb{R}$ and $\phi^2 + \boldsymbol{\epsilon_e}^T\boldsymbol{\epsilon_e} = 1$, expressed with respect to the robot base frame.

Let us consider the operational space formulation of the dynamic model of a rigid robot manipulator in contact with the environment

$$\mathbf{M_x}(\mathbf{q})\dot{\boldsymbol{\upsilon}}_\boldsymbol{e} + \mathbf{C_x}(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\upsilon_e} + \mathbf{g_x}(\mathbf{q}) = \mathbf{h_c} - \mathbf{h_e} \tag{1}$$

where $\mathbf{M_x}(\mathbf{q}) = (\mathbf{J}\mathbf{M}(\mathbf{q})^{-1}\mathbf{J}^T)^{-1} : \mathbb{D} \rightarrow \mathbb{R}^{n\times n}$ is the positive definite inertia matrix, $\mathbf{C_x}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}^{-T}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{J}^{-1} - \mathbf{M_x}(\mathbf{q})\dot{\mathbf{J}}\mathbf{J}^{-1} : \mathbb{D} \times \mathbb{R}^n \rightarrow \mathbb{R}^{n\times n}$ is the matrix including centrifugal and Coriolis effects, and $\mathbf{g_x}(\mathbf{q}) = \mathbf{J}^{-T}\mathbf{g}(\mathbf{q}) : \mathbb{D} \rightarrow \mathbb{R}^n$ is the gravity term. $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ are the corresponding quantities defined in joint space. The vector $\mathbf{h_c} = \mathbf{J}^T\boldsymbol{\tau}$ is the equivalent end-effector wrench corresponding to the input joint torques.

A finite position and orientation displacement of the desired frame Σ_d with respect to the end-effector frame Σ_e produced a total elastic wrench defined in the robot base frame as

$$\mathbf{h_\Delta} = \mathbf{h_t} + \mathbf{h_o} \tag{2}$$

where $\mathbf{h_t}$ is the elastic wrench applied to the end-effector in the presence of a finite position displacement and $\mathbf{h_o}$ is the elastic wrench produced from a finite orientation displacement. Using (2) for the computation of the elastic wrench in the case of an infinitesimal twist displacement $\delta\mathbf{x_{de}^e}$ near the equilibrium, and discarding the high-order infinitesimal terms, yields the linear mapping

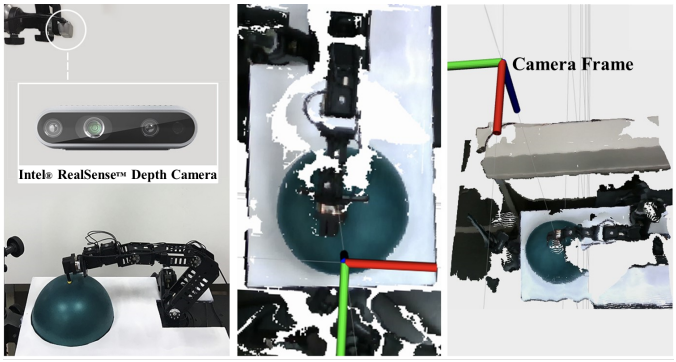
$$\mathbf{h_e^e} = \mathbf{K_P}\delta\mathbf{x_{de}^e} = \begin{pmatrix} \mathbf{K_{Pt}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_{Po}} \end{pmatrix} \delta\mathbf{x_{de}^e} \tag{3}$$

It is apparent that the matrix $\mathbf{K_P}$ represents the stiffness matrix of an ideal spring with respect to a frame Σ_e (coinciding with Σ_d at equilibrium) with the origin at the center of stiffness. Additionally, it can be shown, using (2), that the physical/geometrical meaning of the principal stiffnesses and of the principal axes for the matrices $\mathbf{K_{Pt}}$ and $\mathbf{K_{Po}}$ are preserved also in the case of large displacements. Based on above discussion the active stiffness matrix $\mathbf{K_P}$ can be set in a geometrically consistent way with respect to the task at hand.

Hence, if we denote the end-effector error in the operational space as $\boldsymbol{\Delta}\mathbf{x_{de}} = \mathbf{x_d} - \mathbf{x_e}$ and the corresponding velocity error as $\boldsymbol{\Delta}\dot{\mathbf{x}}_\mathbf{de} = -\dot{\mathbf{x}}_\mathbf{e}$, assuming constant $\mathbf{x_d}$ i.e. considering the surface's deformation as a quasi-static phenomenon, a smooth compliance control with a geometrically consistent active stiffness can be achieved using the control law

$$\mathbf{h_c} = \mathbf{h_\Delta} - \mathbf{K_D}\boldsymbol{\upsilon_e} + \mathbf{g_x}(\mathbf{q}) \tag{4}$$

where $\mathbf{h_\Delta}$ is calculated in (2), $\mathbf{K_D} \in \mathbb{R}^{6\times 6}$ is symmetric, positive-definite and expresses the gain of a D term and $\mathbf{g_x}(\mathbf{q})$ is the gravity compensation term which is known from the robot dynamic model (1). Considering the analysis above, for the problem of robotic interaction control with the surface of a deformable object, it is sufficient to define the desired frame Σ_d, which can be used to define the displacement vectors with respect to the end-effector. Indeed, if the desired frame is defined and assuming that the current end-effector frame is easily determined from the robot kinematics and the measurements on its joints, then the relative pose between the desired and current end-effector frames can be calculated at each time step, along with the corresponding displacement vectors required for the stiffness controller to operate. The desired frame Σ_d can be provided to the controller through the proposed framework described in the following Section.



(a) System Set-up (b) Camera View (c) Occlusion Problem

Fig. 1: A robotic manipulator performing interactive tasks on a hemispherical deformable object. **(a)** The experimental setup used to test the performance of the proposed approach, which includes an Intel®RealSense™depth camera. **(b)** View from the system's depth camera in PointCloud form. **(c)** Tilted view from the system's depth camera in PointCloud form, in which the missing scene information due to occlusions caused by the robot is apparent.

IV. Proposed Framework

In Section IV the proposed interactive motion planning and control framework is presented and its main components are described in detail. A general overview of the overall architecture of the proposed framework is provided in Fig. 2. Starting from the assumption that the initial position of the surface and its deformation evolution (active or passive) over time are unmodelled, a robust perception layer has to be integrated to the robotic system. The visual (RGB-D) information obtained from a camera (Fig. 2 left) is processed and combined with the mesh representation of the object in a sequence of preprocessing steps (Fig. 2 middle), in order to deduce reliably the configuration of the object in the Physical Space, where a stiffness interaction control scheme is implemented (Fig. 2 right-bottom). At the core of the system, a mesh-based version of object deformable surface is constructed and tracked over time in 2D parameterized spaces, where the actual planning of interactive actions is realized (Fig. 2 right-top). The rest of this section presents all the key algorithmic steps and components of the system, describing more in detail the mesh-based paramaterization and associated interactive motion planning phases which constitute the core modules in the proposed framework.

A. Pre-Processing Steps

The initial key step of the whole system is to process, robustly and in real time, rich visual data in order to reliably extract information about the configuration of the deformable target object in the scene. Though this visual pre-processing step does not constitute the focus of this paper, some critical considerations are discussed hereafter, in order to highlight specific constraints and requirements related to the interactive robotic manipulation tasks addressed in this work. A camera can provide rich information about the structure of the surface at high frame rates, however the choice of the camera type is important for coping with some critical data robustness issues. For instance, tracking a physical landmark on the surface is not possible using only the input from a camera, as depicted in Fig. 1(a). Although some approaches have been proposed handling this tracking problem by taking advantage of the image texture [65], [66], these approaches are not suitable for interaction tasks since the visual information may not be available due to the presence of the robot in the scene. More specifically, during the execution of an interaction task, the robot itself might occlude the view of the surface (Fig. 1 (c)) or the nature of the task may modify its visual structure (e.g. polishing), thus disrupting some valuable visual information. On the other hand, the lack of visual features may infer some ambiguities in the process of pose identification of a surface in the 3D space. Therefore, in this work we assume textureless objects, which is more applicable to interaction control tasks, and consequently rely on an RGB-D sensor (Fig. 1 (a)), which can provide 3D visual information and be used to remove the described ambiguities.

Despite the rich 3D visual data obtained from an RGB-D sensor, some critical preprocessing layers of the input data are required. Firstly, an algorithmic layer performing object or region of interest detection and **visual segmentation** is a prerequisite for the purposes of input data reduction and accurate localization of the target object in the manipulation scene. As shown in Fig. 2 the output of this step is a segmented image, including a pixel mask indicating the foreground (i.e. the area of interest in the object's surface) and the background. For each pixel of the foreground, the corresponding depth value from the RGB-D camera is used to initially estimate the shape of the object at each time step. Several algorithms exist in the literature which can be used to perform visual segmentation, such as the ones described in [67], [68], and constitute a major topic in computer vision society. Hence, these algorithms fall outside the scope of this paper and will not be further analysed. Secondly, a **deformation tracking** algorithm, which can combine the visual and depth information from the previous step with the elasticity parameters of the object, constitutes another important preprocessing step. It is important to distinguish deformation tracking from non-rigid reconstruction performed by methods such as those proposed in [69], [11], [70], which reconstruct a single mesh at each frame provided by the vision sensor. This distinction is key to our proposed framework, as will be explained below, since typical non-rigid reconstruction methods provide no guarantee for constant cardinality of the reconstructed mesh elements.

On the other hand, the goal of tracking algorithms is to continuously estimate the deformations and rigid transformations undergone by an object, which is previously modeled by a known mesh. Among the deformation tracking algorithms, which use as an input the Point-Cloud data from an RGB-D sensor, there are some using NURBS parameterization [18], [13] and others using physics-based methods such as discrete mass-spring-damper system [23], [71] or Finite Element Method [48], [24], [25] based on continuum mechanics. These methods register the acquired Point-Cloud data to a mesh with suitable topology as depicted in Fig. 2. Physics-based methods are more suitable for robot-object interaction and manipulation applications, since an explicit physical modelling permits the reliable computation and prediction of internal forces undergone by the object and thus to plan and perform proper force control tasks. Furthermore, the tesselation of the region

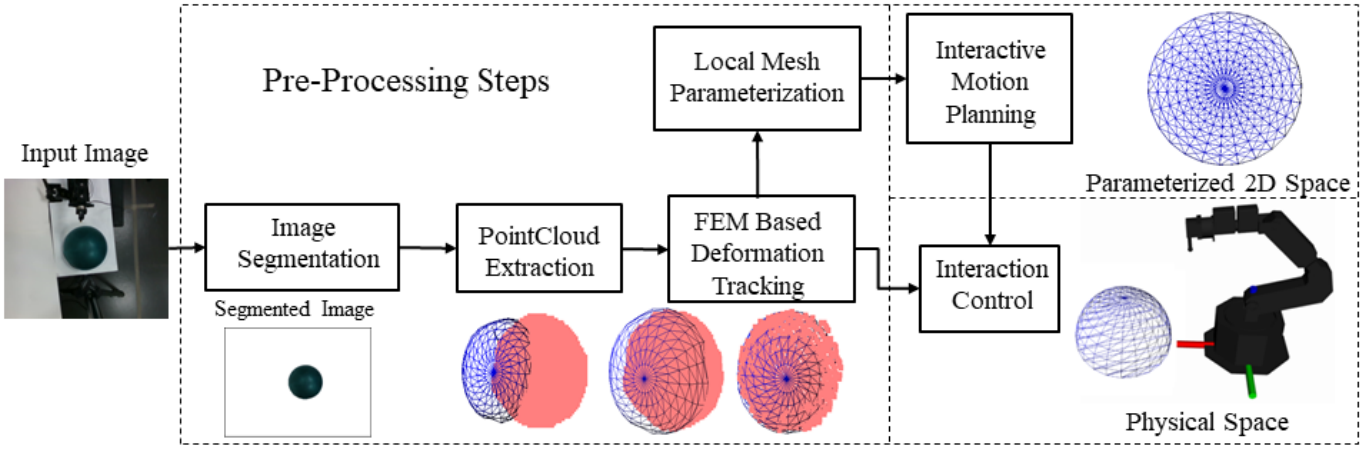


Fig. 2: The overall architecture of the proposed interactive mesh-based motion planning framework. Visual object segmentation, FEM deformation tracking and local mesh parameterization constitute the pre-processing algorithmic steps. These steps provide a mesh representation of the object both in Physical space, where a stiffness interaction control scheme is implemented, and in 2D Parameterized spaces, where planning of interactive actions is realized.

of interest that a mesh provides gives valuable information about the topology of the surface locally, both for normal vector estimation and for calculating geodesic distances between physical points, and is appropriate for the reduction of data noise inserted by the RGB-D sensor. Additionally, the physical simulation process of the tracking algorithms can handle efficiently the visual occlusion problem, especially for the contact area during an interactive task. The position of the mesh elements, which correspond to the occluded areas, can be continuously updated by the physical model using the calculated internal forces, even when the Point-Cloud data are not available, and an estimation of the contact force based on equations (3), (4) and the robot kinematics.

In spite of the accurate deformation estimation achieved with a tracking algorithm at each time frame, the problem of predicting the evolution of the deformation over time remains unsolved. This problem makes the definition of a motion planning strategy impossible, as explained above. Therefore, another pre-processing step is essential; that is, the calculation of a non-linear transformation which will project a set of triangles, representing the area of interaction, to a plane at each time step. We name this planar space as *Dynamic Canonical Space* $\mathcal{W}_{[dc]}$, graphically shown in the middle of Fig. 3. In case we take an instance of this dynamic triangular set at an initial time frame, we can define a static version of the planar space as *Static Canonical Space* $\mathcal{W}_{[sc]}$, depicted in Fig. 3.

An ideal property of this projection transformation concerns distance and angle preservation (isometry), which is though not true for general surfaces, since the Gaussian curvature of an arbitrarily triangulated 3D surface and of a planar surface are not equal. However, extensive literature from computer graphics society exists, [72], [73], [74], proposing a variety of algorithmic approaches for constructing a global close-to-isometric **mesh parameterization**. Most of these approaches are though computationally expensive and thus not suited for real-time planning and control purposes. Nevertheless, local solutions also exist which parameterize a part of the object surface, such as the one presented in [75]. Such solutions have proven both efficient and accurate in terms of isometry, though applied in a completely different context than the one considered in this paper. We employ a local mesh parameterization approach and more specifically the Discrete Exponential Map (DEM) [75], integrated in the core of the proposed interactive motion planning framework, ensuring the imposed performance requirements (bijective transformation properties and computational efficiency), as will be explained in the rest of this Section.

B. Mathematical Formulation

The definition of two different parameterized spaces is useful for tracking differentially the changes of the geodesic distances on the object's surface, coming from the deformation. Hence, at this point, it is useful to present an explicit form of the projection transformation ψ^T which is based on the reference surface in the Physical Space $\mathcal{W}_{[p]}$. Practically, part of the object's surface in Physical Space represents a 2D manifold embedded in $\mathbb{R}^3$, in which the reference surface is expressed as a set of connected triangles $\mathcal{T}_k^{[p]}, k = 1...N$ (Fig. 3), whose vertices are given by the points $\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3 \in \mathcal{W}_{[p]}$. This triangular mesh is the result of the visual segmentation and the deformation tracking preprocessing steps. The above vertices are projected using a mesh parameterization algorithm to points $\mathbf{u}_k^{d1}, \mathbf{u}_k^{d2}, \mathbf{u}_k^{d3} \in \mathcal{W}_{[dc]}$ on the Dynamic Canonical Space (DCS), forming the set of triangles $\mathcal{T}_k^{[dc]}, k = 1...N$. This projection procedure can be expressed as a coordinate transformation ψ, which can be geometrically constructed piecewise inside the $\mathbb{R}^2$ subspace $\mathcal{W}_{[dc]}$ with boundary defined by the parameterized mesh boundary edges.

Let $\mathbf{p}_{[dc]} = (x_{[dc]}, y_{[dc]}) \in \mathcal{T}_k^{[dc]}$ be a point in DCS, which can be expressed in barycentric coordinates (b_1, b_2, b_3) [76] of the triangle, in which it lies, as follows:

$$\mathbf{p}_{[dc]} = b_1\mathbf{u}_k^{d1} + b_2\mathbf{u}_k^{d2} + b_3\mathbf{u}_k^{d3} = [\mathbf{u}_k^{d1}, \mathbf{u}_k^{d2}, \mathbf{u}_k^{d3}] \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (5)$$

with $[\mathbf{u}_k^{d1}, \mathbf{u}_k^{d2}, \mathbf{u}_k^{d3}]$ denoting the edges of the k-th triangle as shown in Fig. 3. Using the property $b_1 + b_2 + b_3 = 1$, (5) can be rewritten in the form

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \mathbf{T}_{[dc]k}^{-1}(\mathbf{p}_{[dc]} - \mathbf{u}_k^{d3}) \quad (6)$$

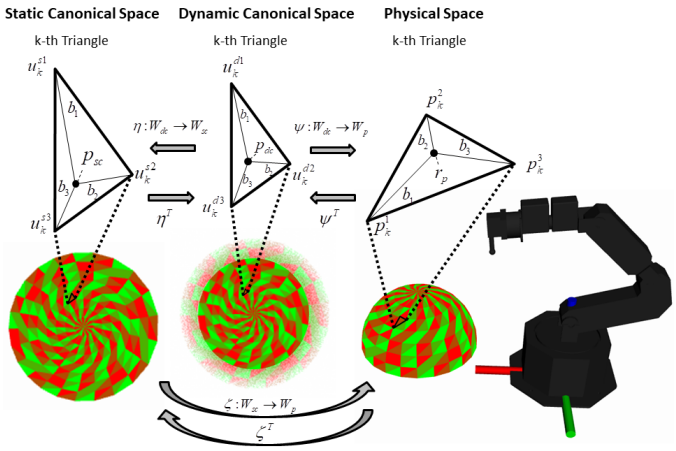


Fig. 3: Establishment of bijective transformations between the Static Canonical Space (SCS), the Dynamic Canonical Space (DCS) and the Physical Space with the use of barycentric coordinates between the corresponding triangles. The position $\mathbf{p}_{[dc]}$ in the k^{th} active triangle of DCS corresponds to the position $\mathbf{p}_{[sc]}$ in the k^{th} active triangle of SCS through the transformation η and simultaneously on the other side to the position $\mathbf{r}_{[p]}$ in the k^{th} active triangle of the Physical Space through the transformation ψ. A direct transformation ζ from the SCS to the Physical Space can also be defined similarly.

where the matrix $\mathbf{T}_{[dc]k} = (\mathbf{u}_k^{d1} - \mathbf{u}_k^{d3},\ \mathbf{u}_k^{d2} - \mathbf{u}_k^{d3}) \in \mathbb{R}^{2\times 2}$ is invertible since the vertices forming a triangle are linearly independent (as long as they are assumed not collinear). The corresponding point $\mathbf{r}_{[p]} \in \mathcal{T}_k^{[p]}$ in the Physical Space is calculated using the function $\psi_k : \mathcal{T}_k^{[dc]} \rightarrow \mathcal{T}_k^{[p]}$

$$\mathbf{r}_{[p]} = \psi_k(\mathbf{p}_{[dc]}) = \mathbf{T}_{[p]k}\mathbf{T}_{[dc]k}^{-1}(\mathbf{p}_{[dc]} - \mathbf{u}_k^{d3}) + \mathbf{p}_k^3 \quad (7)$$

where we define the matrix $\mathbf{T}_{[p]k} = (\mathbf{p}_k^1 - \mathbf{p}_k^3,\ \mathbf{p}_k^2 - \mathbf{p}_k^3) \in \mathbb{R}^{3\times 2}$, with the use of the k-th triangle's edges in Physical Space $\{\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3\}$ (Fig. 3). Extending to the entire DSC we construct the coordinate transformation $\psi : \cup_k \mathcal{T}_k^{[dc]} \rightarrow \cup_k \mathcal{T}_k^{[p]}$ as follows:

$$\mathbf{r}_{[p]} = \psi(\mathbf{p}_{[dc]}) = \sum_{k=1}^{N} (\mathbf{T}_{[p]k}\mathbf{T}_{[dc]k}^{-1}(\mathbf{p}_{[dc]} - \mathbf{u}_k^{d3}) + \mathbf{p}_k^3)\xi_k(\mathbf{p}_{[dc]}) \quad (8)$$

where the selection function $\xi_k(\mathbf{p}_{[dc]})$ is defined as follows:

$$\xi_k(\mathbf{p}_{[dc]}) = \begin{cases} 1 & \text{if } \mathbf{p}_{[dc]} \in \mathcal{T}_k^{[dc]} \\ 0 & \text{elsewhere} \end{cases} \quad (9)$$

Working similarly in Physical Space, we can express $\mathbf{r}_{[p]}$ barycentric coordinates (b_1, b_2, b_3) of the k-th triangle, in which it lies in the Physical Space, as follows:

$$\mathbf{r}_{[p]} = b_1\mathbf{p}_k^1 + b_2\mathbf{p}_k^2 + b_3\mathbf{p}_k^3 = [\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3] \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (10)$$

Since the matrix $[\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3] \in \mathbb{R}^{3\times 3}$ is invertible, we can rewrite (10) in the following form:

$$[b_1, b_2, b_3]^T = [\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3]^{-1}\mathbf{r}_{[p]} \quad (11)$$

Substituting (11) into (5) we can rewrite (5) as follows:

$$\mathbf{p}_{[dc]} = [\mathbf{u}_k^{d1}, \mathbf{u}_k^{d2}, \mathbf{u}_k^{d3}][\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3]^{-1}\mathbf{r}_{[p]} \quad (12)$$

Extending to the entire Physical Space we can construct the coordinate transformation $\psi^{-1} : \cup_k \mathcal{T}_k^{[p]} \rightarrow \cup_k \mathcal{T}_k^{[dc]}$ as follows:

$$\mathbf{p}_{[dc]} = \psi^{-1}(\mathbf{r}_{[p]}) = \sum_{k=1}^{N} [\mathbf{u}_k^{d1}, \mathbf{u}_k^{d2}, \mathbf{u}_k^{d3}] \cdot [\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3]^{-1} \cdot \mathbf{r}_{[p]} \cdot \xi_k(\mathbf{r}_{[p]}) \quad (13)$$

where the selection function $\xi_k(\mathbf{r}_{[p]})$ is defined as follows:

$$\xi_k(\mathbf{r}_{[p]}) = \begin{cases} 1 & \text{if } \mathbf{r}_{[p]} \in \mathcal{T}_k^{[p]} \\ 0 & \text{elsewhere} \end{cases} \quad (14)$$

It is noteworthy that the summation of equations (8) and (13) is reduced to a single term, since $\mathbf{p}_{[dc]}$ and $\mathbf{r}_{[p]}$ respectively can lie on only one triangle, denoted as **active triangle**. In the edges and vertices which constitute the boundaries between two triangles the active triangle can easily be chosen with the use of a heuristic function. Additionally, the use of the deformation tracking algorithm described above allows for constant cardinality of triangles in the mesh structure across all the time frames, which is important for bijectivity preservation of the transformation ψ and continuity of the planned interactive tasks on the surface of the object. We can follow the same procedure to calculate a bijective coordinate transformation $\zeta : \mathcal{W}_{[sc]} \rightarrow \mathcal{W}_{[p]}$ between the Static Canonical Space (SCS) and the Physical Space (not presented here for brevity) and the bijective transformation η between $\mathcal{W}_{[sc]}$, which is a subspace of $\mathbb{R}^2$ with boundary defined by the boundary edges of static parameterized mesh (Fig. 3), and $\mathcal{W}_{[dc]}$ defined above.

It should thus be highlighted that the use of barycentric coordinates and the parametrization of the mesh structures as defined in our approach, are crucial for the bijectivity proof of the transformations needed to enable appropriate planning of the interactive tasks considered in this work. In particular, the barycentric coordinates allow us to navigate in the interior of each triangle on the mesh structures and prove that the Physical space is bijective to the Dynamic and Static physical spaces with constant triangle cardinality at each time step. Similar bijectivity properties have been proven for mesh structures in [77], however the authors have made the restrictive assumption that the surface should have a terrain form (or 2.5D map), which is not appropriate for deformable objects. Comparing to what is described in [77], our approach mainly differentiates on the definition of the canonical space, which is based on mesh parameterization as described in the last two paragraphs of Section IV-A. This is a crucial pre-processing step to our framework, since it allows us to generalize to more complex areas of deformable surfaces, in which the terrain form assumption is not valid, and derive a complete bijectivity proof effectively inverting the transformation as described above.

C. Interactive Motion-Planning Approach

The core of our proposed framework for surface interaction control tasks is based on an online calculation of the desired frame Σ_d, which is used as an input to the stiffness controller,

as described in Section III. A central idea in this framework is to make use of the preprocessing steps, described in the previous paragraphs, in order to firstly construct the spaces denoted as Physical and its planar parameterized versions denoted as Static Canonical and Dynamic Canonical. Through the respective coordinate transformations between these spaces, computed as defined above, it then becomes possible to continuously update the position and orientation of the desired stiffness control frame Σ_d at each time step.

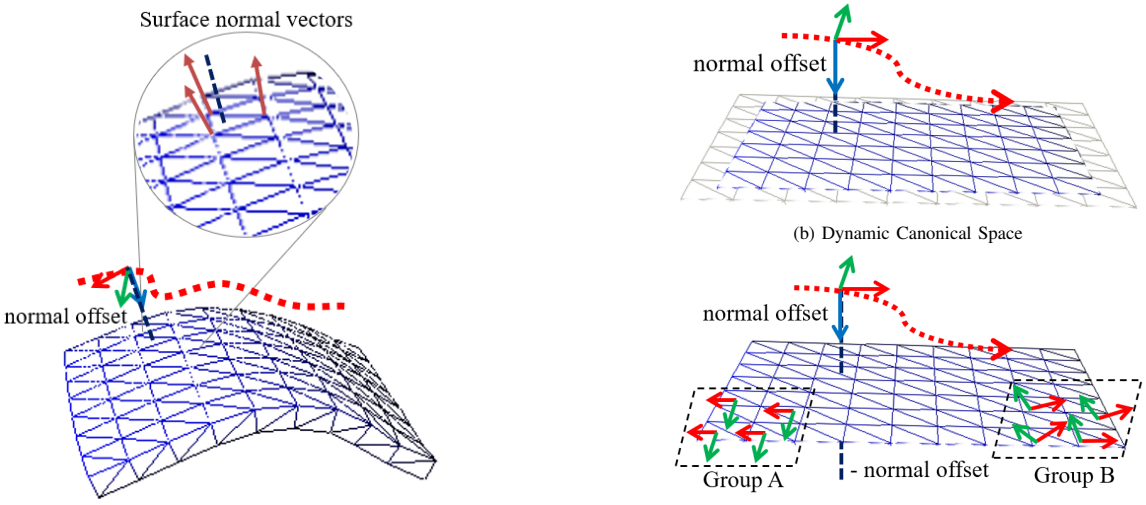
Starting from the calculation of the position part in the desired control frame, the goal would be to achieve both motion planning on the interaction area and regulation of the interaction force. The first sub-goal can be accomplished by planning the trajectories in the Static Canonical Space. We take advantage of its 2D static nature and structure, in order to plan paths suitable for the interaction task, as shown in Fig. 4c. The planning procedure can be realized within the boundary of SCS, using well-known path planning algorithms [78] with integrated obstacle avoidance techniques, if some areas on the surface have to be avoided during the robotic interaction. Planning of the interaction trajectory is then possible by sampling the paths at proper time steps. At each time step, the sampled point $\mathbf{p}_{[sc]}$ from the path will lie inside a triangle from the set $\mathcal{T}_k^{[sc]}$, defining the active triangle as described above and marked with green color in Fig. 4c. With the use of transformation $\zeta : \mathcal{W}_{[sc]} \rightarrow \mathcal{W}_{[p]}$ we are able to calculate the corresponding point of the Physical Space. Hence, the interaction trajectory can be expressed in the Physical Space in real-time, since the computational cost of the transformation calculations is low, requiring only the calculation of barrycentric coordinates.

However, the above strategy is sufficient for planning trajectories only on the reference surface, which would be sufficient only for contact tasks. For regulation of the contact force and planning of non-contact tasks, we add a vertical offset in the planning procedure, which is defined differently between the coordinate spaces. Namely, in SCS the offset ($z_{[sc]}$) is set along the z-axis of its inertial coordinate frame (Fig. 4c), whereas in Physical Space it is set along the direction of the normal vector of the active triangle, whose direction is calculated with the use of the local barycentric coordinates, as depicted in Fig. 4a. More specifically for the directional part, the mesh's normal vectors are calculated in its edges at each time frame by the mesh tracking algorithm as a linear combination of the neighbouring triangles' normals. Hence, the barrycentric coordinates calculated by the transformation $\zeta : \mathcal{W}_{[sc]} \rightarrow \mathcal{W}_p$ also provide a linear combination of the mesh's normals at each point inside a triangle, implementing a normal shading algorithm. In terms of offset values, for positive ones ($z_{[sc]} > 0$), the desired position is translated along the direction of the normal vector above the surface, thus resulting in a non-contact interaction task between the robot and the surface. In case of negative offset values, the desired position is set below the surface level, achieving contact with regulated interaction force by adjusting the offset value and hence the translational displacement error. By planning smooth trajectories including the vertical offset, the robot can navigate across the whole area of interest, with desired motion and contact force properties defined by the interaction trajectory.

Algorithm 1: Interactive Motion Planning

SCS : Static Canonical Space (2D Mesh)
DCS : Dynamic Canonical Space (2D Mesh)
PS : Physical Space (3D Mesh)
PCL : Point Cloud
$Bary$: Barycentric Coordinates
AT : Active Triangle
$PProp$: Physical Properties
$InputIm$: Input Image
$SegIm$: Segmented Image
$DepIm$: Depth Image
$SegIm$ = **Visual Segmentation**($InputIm$)
PCL = **Calculate PointCloud**($SegIm, DepIm$)
PS = **Deformation Tracking**($PCL, PProp$)
SCS = **Mesh Parameterization**(PS)
$DCS = SCS$
while *Not End of Desired Trajectory* **do**
  $\mathbf{q}$: Actuation Vector
  $\mathbf{p}_{[dc]}$: Trajectory Point in DCS
  $\mathbf{r}_{[p]}$: Trajectory Point in PS
  $\mathbf{o}_{[p]}$: Desired orientation in PS
  Σ_d : Desired Frame
  Σ_e : End-Effector Frame
  $\mathbf{h_e}$: End-Effector wrench
  VO : Vertical Offset
  $TV_{[dc]}$: Tangent Vectors in DCS
  AT = **Compute Active Triangle**($\mathbf{p}_{[dc]}, DCS$)
  $Bary$ = **Compute Barycentric**($\mathbf{p}_{[dc]}, AT$)
  VO = **Compute Vertical Offset**($\mathbf{p}_{[dc]}$)
  $\mathbf{r}_{[p]}$ = **Compute Position**($PS, AT, Bary, VO$)
  $\mathbf{o}_{[p]}$ = **Compute Orientation**($PS, AT, Bary, TV_{[dc]}$)
  $\Sigma_d = [\mathbf{r}_{[p]}, \mathbf{o}_{[p]}]$
  Σ_e = **Robot Kinematics**($\mathbf{q}$)
  $\mathbf{h_e}$ = **Stiffness Controller**($\Sigma_d, \Sigma_e, \mathbf{q}$) (Eq. 4)
  Compute and execute motor torques
  $SegIm$ = **Visual Segmentation**($InputIm$)
  PCL = **Calculate PointCloud**($SegIm, DepIm$)
  PS = **Deformation Tracking**(PCL, P_Prop)
  DCS = **Mesh Parameterization**(PS)
end

It is noteworthy that the geodesic distances on the surface of an object, which undergoes elastic deformations, are changing. This means in practice that the length of one step along a path planned in SCS is different than the corresponding step in Physical Space. It is obvious that, if the requirements of the interaction task include preservation of distance from a landmark point or of constant velocity on the surface, the motion planning strategy defined above is not sufficient. In this case, a different planning strategy has to be adopted, which involves the Dynamic Canonical Space Fig. 4b. In particular, we keep on planning the same interaction paths, but inside the DCS in this case, and then perform sampling



(a) Physical Space

(b) Dynamic Canonical Space

(c) Static Canonical Space

Fig. 4: The proposed mesh-based interactive motion planning approach. **(a)** The result of a deformation tracking preprocessing step is the mesh model of the object fitted on the Pointcloud data obtained by the depth camera. The vertices of each triangle, where the local normal vector is defined, provide information about the topology of the surface locally. The planned path is fitted on the mesh surface depicted as red dots. The reference pose translated along the normal offset is depicted as a reference frame with a blue normal vector and green-red tangential vectors, and is calculated with normal shading techniques based on barycentric coordinates. **(b)** The flattened mesh surface is the result of the parameterization preprocessing step. For each time step, the parameterization forming the Dynamic Canonical Space is calculated, which tracks the geodesic changes of the object's mesh model. If the object is not actively deforming, the DCS equals to the SCS. **(c)** At an initial time step, the parameterized Static Canonical Space is obtained, in which the robot's reference motions are planned (depicted as a red path). The tangential directions can be computed axis aligned (Group A) or can follow the direction of the principal directions (Group B). Each planned reference pose is depicted with a reference frame with the blue vector representing the normal direction and the green-red the tangential directions. The planning in the parameterized space includes a normal offset w.r.t. the surface of the mesh.

along that path at proper time steps, for interaction trajectory planning according to the task requirements. As presented in Algorithm 1, the sampled point $\mathbf{p}_{[dc]}$ from the path will lie inside a different triangle as compared to the SCS, defining a new active triangle, which is marked with green color in Fig. 4b. The continuous surface parameterization implemented with a close to isometric surface parameterization algorithm allows for continuous monitoring of the changes in the surface geodesic distances in DCS. In this way, the planned trajectories in DCS are then transformed to the Physical Space with the use of transformation $\psi : \mathcal{W}_{[dc]} \rightarrow \mathcal{W}_{[p]}$, incorporating the distance modifications imposed by the object's deformation. Moreover, the vertical offset will be set in DCS along the z-axis of its inertial frame, similarly to SCS. It is important to mention that in terms of computational efficiency the proposed transformations between the Canonical and Physical spaces are essentially based on a set of linear computations and one-to-one correspondences, which can be executed in real-time.

Regarding the calculation of the orientation part, we commence with the calculation of the surface normal vector based on a normal shading technique. More specifically, as shown in Fig. 4a the vertices of each triangle, where the local normal vector is defined, provide information about the topology of the surface locally. This topological information is exploited for the continuous calculation of the local normal vector with the use of barycentric coordinates with the relation $\mathbf{n} = b_1\mathbf{n}_1 + b_2\mathbf{n}_2 + b_3\mathbf{n}_3$, where $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$ are the normal vectors defined on the vertices of the active triangle and b_1, b_2, b_3 are the barycentric coordinates of the target point on the surface as described in Sec. IV-B. On the boundary between two triangles the summation contains only the common vertices of the triangles. Besides computing the local normal vector, it is also necessary to define appropriately the tangent vectors. Since the path planning is implemented in SCS, we can set the reference tangent vectors of each triangle representing the local x and y axes, according to the planning strategy. For example, we can choose to set all the tangent vectors aligned to the SCS inertial frame, indicated in Fig. 4c as Group A, or to set them along the principal directions (Group B in Fig. 4c), calculated locally from the reference surface at the time frame the SCS was captured. It is important, for the continuity of the planned motions, that the direction of reference tangent vectors in neighbouring triangles does not assume large deviations. The reason for this is that the orientation part of the desired frame Σ_d is calculated relatively to the local reference frame of the active triangle, formed by the reference tangent vectors and the vertical direction of each space (i.e., the z-axis of SCS and DCS and the local normal vector in the Physical Space). Similarly to the position calculation, we plan the desired orientation in the planar spaces with respect to the local reference frame. By using the proper transformations, we then calculate the corresponding reference frame in the Physical Space, which consequently defines the desired orientation of the stiffness control frame Σ_d and is used for the calculation of the orientation displacement error, as described in Sec.III.

V. Experimental Evaluation

The goal of the conducted experiments is to assess the performance of the proposed approach in several surface interaction tasks with a deformable object. The performance criteria include interaction trajectory tracking accuracy, computational efficiency and contact force regulation. These criteria drive our decision for choosing proper implementations of the preprocessing steps and integrate them to the overall system.

A. Pre-processing Steps

The visual segmentation preprocessing step is used as an input to the deformation tracking algorithm, in order to detect the object of interest on the image plane and to reduce the number of points in the acquired Point-Cloud. For the implementation of this step, we employ the popular GrabCut method, implemented efficiently in GPU, described in [79]. This algorithm addresses the segmentation task as an energy minimization problem based on statistical models of the foreground and the background. For the deformation tracking step, physics based algorithms, are more suitable for robotic interaction tasks, as described in Section IV. For this experimental validation we employ the FEM based method proposed in [24], [25], because it provides more accurate deformation tracking for large elastic deformations, as compared to mass-spring-damper approaches, with real-time computational performance ($\sim$35fps). Moreover, this method relies on a volumetric linear FEM approach, which computes the deformation field over the elements of a known tetrahedral mesh approximating the object of interest. The use of tetrahedrons is convenient for meshing volumes with high complexity, since it offers accurate modelling and computational efficiency at the same time. Regarding the mesh parameterization, we integrated into our system the Discrete Exponential Map (DEM) presented in [75]. DEM is a local parameterization algorithm, which focuses on the computation of normal coordinates of a discrete surface (mesh) using the Dijkstra's algorithm with very high computational efficiency ($\sim$100 fps).

B. Experimental Set-Up

In order to test and analyze the performance of the proposed framework, an experimental set-up is used which includes an Intel® RealSense™ depth camera and a Widow-XL robotic manipulator by Trossen Robotics to perform the interactive tasks. The Widow-XL robot is a 5-DoF manipulator equipped with Dynamixel MX Series Servo motors operated in torque control mode. The geometrically consistent active stiffness controller briefly described in Sec.III-A is implemented to control the motion of the robot, running at 200Hz. One of the goals of this evaluation procedure is to quantitatively measure the accuracy of motion planning in interactive tasks (both contact and non-contact), using our proposed framework, and of its respective execution by the robot. The experiments involve interactive motion tasks over an object of interest both in a stationary scenario and in a deforming scenario. In order for the generation of ground truth data to be feasible, the object's shape is required to be known at each time step, especially for the deforming scenarios. Therefore, in this experimental study, we chose to use a hemispherical model, which can undergo uniform deformations and its center (termed ground truth frame) has a known relative position and orientation with respect to the robot's base frame (Fig. 5).

The hemispherical model is realized with the upper part of a balloon (see Fig. 1). The shape of the balloon model is controlled with the use of a deformation control mechanism specifically constructed for this experimental procedure. The air flow which affects the shape of the balloon model is

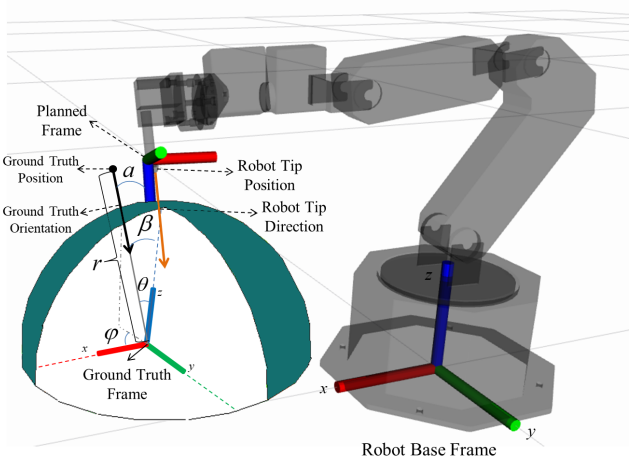


Fig. 5: Illustration of the experimental set-up: Hemispherical model used with known relative position and orientation with respect to the base frame of the robot manipulator. This model is used to derive ground-truth data for evaluating position and orientation accuracy of both the planned motion and the motion actually executed by the robotic manipulator.

provided by an air pump. The DC motor of the air pump is controlled by a PWM speed controller, modifying the resistance of its potentiometer. The potentiometer value at each time step is controlled by an algorithm executed in a Raspberry Pi 3 micro-computer, with the use of a servo motor.

The angle of the servo motor linearly modifies the resistance value of the potentiometer, which modulates the PWM signal of the speed controller. With accurate speed control of the air pump's motor, we are able to control the air flow inserted to the balloon model through a tube. However as the balloon's pressure is increasing and becomes larger than one bar, an outgoing air current is formed which flows from the balloon to the atmosphere. Consequently, the pneumatic system finds its equilibrium and the balloons shape doesn't change. By controlling the speed of the air pump, we are able not only to control the shape of the balloon model but also its deformation rate. The relative position of the center of the hemispherical model that is formed by the upper part of the balloon model with respect to the robot base frame is measured accurately as: $x_c = 0.26m, y_c = 0.025m, z_c = 0.08m$, denoted as Ground Truth Frame in Fig. 5, aligned with the Robot Base Frame.

C. Experimental procedures and results

*1) **Interactive Motion Planning Accuracy**:* We start by expressing each interactive task in spherical coordinates following the ISO convention, i.e. denoting $(r, \phi, \theta) : r \geq 0, 0° \leq \phi \leq 360°, 0° \leq \theta \leq 180°$, as the radius, the azimuthal angle and the polar angle, respectively. The ground truth orientation is set, without loss of generality, to be perpendicular to the surface of the hemispherical model, adding a 2DoF motion constraint to the position constraints of each interactive task, to conform to the 5 DoF kinematic structure of the available robot. Hence, it is calculated as the orientation of the vector starting from the ground truth position and passing through the model's center position, depicted with a black arrow in Fig. 5. The remaining orientation component is set in a way that is always reachable by the robot. The relative planned and executed robot's position is calculated

with respect to the hemisphere's center and then expressed in spherical coordinates for comparison with ground truth data. The use of spherical coordinates allows for accuracy assessment along both the perpendicular and the tangential directions of the object's surface, enabling in particular a better characterisation of the performance in relation to the mesh parameterization algorithm, which constitutes a key element in the process. Considering comparison in terms of the orientation data, the angle α between the z-axis of the planned reference frame (blue vector in Fig. 5) and the ground truth orientation is calculated at each time step. Similarly for the robot's orientation error, the angle β between the z-axis of the robot's tip frame (orange arrow in Fig. 5) and the ground truth orientation is computed as well. Thus, with the proposed experimental protocol we are able to measure both the position and orientation errors and to properly evaluate the interactive motion planning and execution accuracy. For the deforming scenarios, the balloon model was periodically inflated and deflated between the spherical radius values $R_{\max} = 11\,\text{cm}$ and $R_{\min} = 10\,\text{cm}$, using two different deformation rates of $0.25\,\text{Hz}$ and $0.5\,\text{Hz}$, in two distinct experimental conditions. The choice of 1 cm deformation was mainly imposed by hardware and experimental set-up limitations, in a way that (i) the balloon could maintain its hemispherical shape and its deformation could be accurately controlled and (ii) the robot could operate within its limited dexterous workspace. Nevertheless, this deformation represents a change of approx. 20% in the surface of the target hemispherical area, which is sufficient to assess the performance of the method, as described in the sequel.

To assess the performance of the proposed framework in a set of indicative experimental conditions, two different interactive tasks were planned and executed by the robotic manipulator in both stationary and deforming scenarios. More specifically, the first interactive path is a **circular path** with geodesic radius $2.5\,\text{cm}$ away from the pole, shown in Fig. 6. The second interactive path is a **combined path**, comprised both of circular (with a geodesic radius of $2.5\,\text{cm}$) and meridian paths (of different azimuthal angles), as shown in detail in Fig. 7. The selection of each path aims to validate the performance of the framework in terms of geodesic distance preservation for interactive tasks. Using as reference the polar point of the hemispherical model, the circular path validates the ability of our approach to preserve constant geodesic distance from a specific point despite the deformation of the surface, while through the meridian parts in the combined path we aim to quantify the ability to control the velocity of the robot on the object's surface. In addition, the combined path provides insights on the performance of our framework on a motion with abrupt direction changes. The path requirements regarding the geodesic distance from the pole of the hemispherical model are set according to the robot kinematic constraints, i.e. so that the robot can remain in its dexterous workspace. They are highly important for this experimental evaluation, since they have to be respected both for the stationary and the shape deformation scenarios. As it can be deduced from the proposed evaluation protocol, the value of the polar angle θ is indicative of the geodesic distance from the

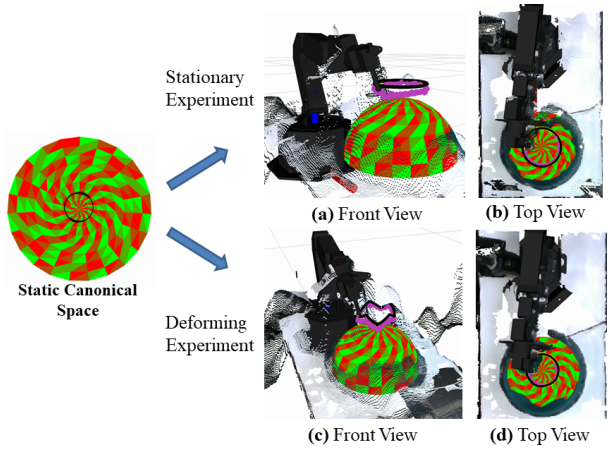


Fig. 6: Circular path execution in **(a) & (b):** a stationary scenario and **(c) & (d):** a deforming scenario. The interactive task is planned in the Static Canonical Space (depicted with a black colored path) and is transformed to the Physical Space with the use of the proposed framework (also depicted with a black colored path). The magenta path depicts the actual interactive task executed by the robot. (a) Front view of the stationary experiment. (b) Top view of the stationary experiment. (c) Front view of the deforming experiment. (d) Top view of the deforming experiment.

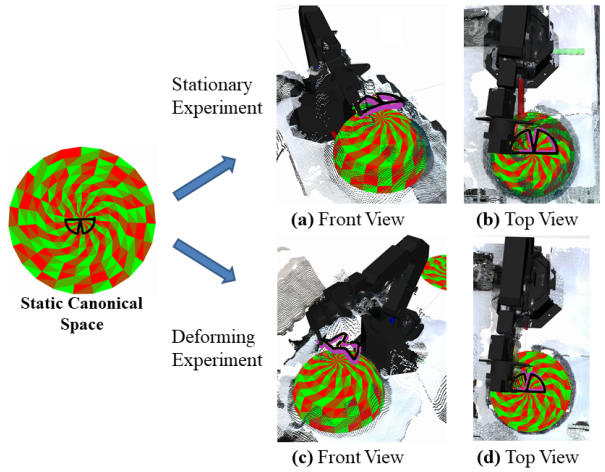


Fig. 7: Combined path execution in **(a) & (b):** a stationary scenario and **(c) & (d):** a deforming scenario. The interactive task is planned in the Static Canonical Space (depicted with a black colored path) and is transformed to the Physical Space with the use of the proposed framework (also depicted with a black colored path). The magenta path depicts the actual interactive task executed by the robot. (a) Front view of the stationary experiment. (b) Top view of the stationary experiment. (c) Side view of the deforming experiment. (d) Top view of the deforming experiment.

polar point, since these two values are linearly dependent with the sphere radius R. For all the tasks in these experiments, the interactive motion planning goal is to keep a constant perpendicular distance of $3\,\text{cm}$ from the surface of the model.

Figures 6 and 7 present an overview of indicative results obtained, regarding the planning and execution of interactive paths, for the two experimental scenarios considered in this study. The paths are planned in the Static Canonical frame following the requirements mentioned above. The proposed interactive motion planning framework achieves accurate transformation of the planned paths to the robot's Physical Space despite the visual occlusion caused by the presence of the robot in the scene, as shown by the black colored paths in the Figures. By comparing the paths executed in the stationary and deforming conditions, as depicted by the front views of

TABLE I: Planning and execution accuracy errors for two interactive trajectories (Circular, Combined), expressed in spherical coordinates, r (cm), θ (rad$\times 10^{-2}$), ϕ (rad $\times 10^{-2}$) and orientation errors for planning α (rad $\times 10^{-2}$) and robotic execution β (rad $\times 10^{-2}$), for the stationary object scenario with spherical radius values $R_{\max} = 11$ cm and $R_{\min} = 10$ cm and the deforming object scenario with deformation rate $f = 0.25$ Hz & $f = 0.5$ Hz. The planning errors of the current interactive mesh approach (denoted as **IMP**) are compared to the planning errors of our previous direct visual planning approach (denoted as **DVP**) and to the planning errors of our previous direct visual planning approach with occlusion from the robot motion (denoted as **DVPO**). The robot execution errors (i.e. evaluated on the path actually executed by the robotic manipulator) are denoted as **Robot**.

Planning		**Stationary Circular**		**Deforming Circular**		**Stationary Combined**		**Deforming Combined**	
Accuracy		$R = 10$cm	$R = 11$cm	$f = 0.25$ Hz	$f = 0.5$ Hz	$R = 10$cm	$R = 11$cm	$f = 0.25$ Hz	$f = 0.5$ Hz
$\mathbf{r}$	IMP	0.3 | 0.3±0.1	0.5 | 0.5±0.1	0.3 | 0.3±0.1	0.3 | 0.3±0.1	0.2 | 0.2±0.1	0.5 | 0.5±0.1	0.3 | 0.3±0.1	0.3 | 0.3±0.1
	DVP	0.3 | 0.3±0.3	0.5 | 0.2±0.7	0.8 | 0.4±0.8	1.3 | 0.7±0.6	0.3 | 0.5±0.4	0.3 | 0.2±0.5	0.7 | -0.7±0.4	1.0 | -0.7±0.2
	DVPO	2.4 | 0.6±2.3	2.5 | 1.6±1.9	2.7 | 0.9±2.8	2.8 | 1.9±2.5	1.2 | 0.8±0.9	1.0 | 0.5±0.7	1.4 | 0.9±1.0	1.4 | 0.7±0.7
	Robot	1.0 | 0.6±0.3	1.1 | 0.8±0.3	1.0 | 1.0±0.3	1.0 | 1.0±0.3	0.9 | 0.6±0.2	1.1 | 0.8±0.2	1.0 | 0.9±0.3	1.0 | 0.9±0.4
$\boldsymbol{\theta}$	IMP	1.3 | -1.2±0.3	1.2 | -1.7±0.3	1.2 | -1.1±0.6	1.2 | -1.2±0.3	1.3 | -1.2±0.6	1.4 | -1.5±0.7	1.3 | -1.1±0.6	1.3 | -1.1±0.6
	DVP	3.8 | -3.1±2.1	3.5 | 1.2±2.2	4.6 | -2.3±4.3	7.5 | 3.2±3.4	4.4 | -1.1±5.3	3.8 | 1.0±2.6	6.0 | -0.3±3.3	8.1 | -0.5±3.7
	DVPO	9.3 | 1.6±9.2	10.1 | 2.6±9.8	9.7 | 1.6±8.6	10.5 | 2.8±8.3	12.5 | -3.1±12	12.2 | 0.6±12	13.0 | -3.6±12	13.3 | 2.2±13
	Robot	3.9 | -2.9±0.9	3.5 | -3.2±0.8	2.8 | -2.5±2.9	3.0 | -2.8±1.0	4.3 | -2.6±3.0	4.6 | -2.8±2.7	3.8 | -2.5±2.9	3.8 | -2.5±2.9
$\boldsymbol{\phi}$	IMP	1.8 | -0.6±1.0	1.2 | -0.5±1.0	3.0 | -0.3±1.6	2.7 | 0.5±1.0	1.8 | 1.2±1.4	1.9 | 1.2±1.5	1.6 | 0.4±1.2	1.6 | -0.2±1.6
	DVP	7.6 | -0.7±4.7	7.0 | -1.5±6.0	10.6 | -2.8±8.2	9.0 | 0.9±8.9	6.1 | 2.8±7.1	7.4 | 4.2±7.7	7.0 | 0.8±8.3	7.2 | 0.8±8.3
	DVPO	45.9 | 24±47	51.5 | 22±56	47.8 | 29±48	51.9 | 26±68	73.5 | 54±64	71.0 | 39±71	74.2 | 57±64	73.2 | 34±72
	Robot	3.9 | 0.1±2.9	3.5 | 0.6±2.4	5.0 | 0.9±5.9	6.5 | -5.2±3.9	4.3 | 0.4±2.2	4.6 | 0.6±2.6	6.0 | 1.4±5.9	6.0 | 2.1±8.2
$\boldsymbol{\alpha}$	IMP	2.7 | 3.7±0.9	2.2 | 4.1±0.6	2.5 | 3.5±1.2	2.5 | 3.6±1.4	3.0 | 2.8±1.0	3.1 | 3.1±1.2	3.0 | 2.8±1.0	3.0 | 2.8±1.0
	DVP	5.6 | 4.4±3.5	5.8 | 3.4±7.1	5.7 | 6.4±6.3	6.1 | 4.4±7.6	5.4 | 4.6±2.8	4.7 | 2.8±2.4	8.8 | 3.5±2.5	9.7 | 3.7±3.2
	DVPO	24.6 | 23±8.1	25.0 | 24±6.9	24.6 | 26±8.6	25.6 | 25±7.1	21.8 | 20±8.2	20.6 | 20±6.4	22.6 | 21±8.7	21.8 | 22±5.1
$\boldsymbol{\beta}$	Robot	3.8 | 3.7±1.0	3.6 | 3.4±1.1	3.7 | 4.7±6.8	3.9 | 4.6±6.5	6.1 | 5.5±2.5	5.5 | 5.1±2.2	5.6 | 5.2±2.2	5.7 | 5.3±2.3

Data presented as RMSE relative to ground truth data and mean ± standard deviation of the error values are presented after the vertical bar

the experimental scene shown in the Figures, one can clearly observe the motion adaptation performed to compensate for the surface deformation. Furthermore, the apparent path similarity between the stationary and the deforming scenarios, as shown particularly in the top views of these Figures, visually demonstrates that, by using the proposed Dynamic Canonical Space, the planning procedure actually manages to properly compensate for the object deformation, despite the presence of large visual occlusions.

For the purposes of a quantitative performance evaluation, we conducted a comparative study between the proposed interactive framework and our previous work [80], where a planning algorithm was proposed which uses directly the Point-Cloud data of an RGB-D Camera. The errors in position (measured in spherical coordinates) and orientation are presented comparatively in Table I, for the stationary and the deforming experiment, respectively. For each accuracy metric, the errors presented in the Table are measured with respect to the ground truth position and orientation data (as explained in Section V-B). Table I presents errors characterising the accuracy of motion planning in three different experimental conditions: a) using the interactive mesh approach for motion planning proposed in this paper (denoted as IMP), with the robot (and the large induced occlusions) present in the scene, b) using the direct visual approach for planning from our previous work (denoted as DVP), without occlusions in the scene, and c) using the previous DVP approach, further subject to occlusions from the robot (denoted as DVPO). The Table also present results regarding the accuracy of the actual motion executed by the robot when using our proposed IMP approach (actual motion execution errors are denoted as 'Robot' in the Table). The data presented in the Tables show RMSE values evaluated relative to ground truth data, as well as as mean and standard deviation values of the errors measured along the whole duration of the interactive tasks (with the mean values denoting a positive or negative error bias during motion and the standard deviation values denoting actual error fluctuations indicating precision of motion).

Analysing more in detail the performance of the proposed IMP approach, as presented in Tables I, it can be observed that the maximum RMSE value for the radius (r), both in the stationary and in the deforming scenarios, is 0.5 cm, while the maximum RMSE value for the polar angle (θ) is 1.4×10^{-2} rad in the combined path of the stationary experiment with $R = 11$ cm (resulting in 0.16 cm geodesic distance error). It is evident that the planned paths follow the requirements of the tasks, with sub-centimeter accuracy, both in terms of the relative radial distance and the geodesic distance. Evaluating comparatively the RMSE results between the stationary and the deforming scenarios, it is clear that the changes both in radial and geodesic distances during the evolution of the deformation are effectively compensated. The orientation error α of the planned motion with respect to the ground truth has the largest value of 3.0×10^{-2} rad (or $1.75°$) in the circular deforming interactive task (in the case of the higher frequency of 0.5 Hz), maintaining perpendicular orientation to the hemispherical model's surface. The errors in azimuthal angle (ϕ) are indicative of the projection accuracy of each interactive task into the Physical Space, including its timing constraints. The largest RMS error recorded for the planned motion is 3.0×10^{-2} rad in the circular path deforming experiment, resulting in 0.33 cm geodesic distance

error (estimated for $R_{\max}$). It is evident that the timing constraints of the task are respected when using the proposed IMP framework. The larger values of ϕ in standard deviation comparing to the other position accuracy metrics derive from the large steps of the ISO convention followed in this analysis. Using the planned motion at each time step, the robot is able to execute all the interactive tasks with a precision which is subject to its kinematic (with limited dexterous workspace) and joint control constraints. The implemented stiffness controller, described in Section III-A, has proven convergence in the desired position for non-contact scenarios. However, the choice of its gains directly affects the robot's precision. More specifically, the quantitative analysis presented in Table I shows that the maximum RMSE value for the radial motion (r) (for all stationary and deforming scenarios) is 1.1 cm (recorded in the stationary circular condition) and the maximum RMSE value for the polar angle (θ) is 4.6×10^{-2} rad (recorded in the stationary combined scenario), resulting in 0.50 cm geodesic distance error. An important observation is that the robot exhibits similar behavior both in the stationary and the deforming scenarios, managing to compensate for a deformation of different frequencies between the spherical radius values $R_{\max}$ and $R_{\min}$. The orientation error β of the robot motion relative to the ground truth has the largest RMSE value of 7.6×10^{-2} rad (or $4.37°$) in the circular deforming interactive task, deriving mainly from the robot kinematic constraints due to its 5 DoF structure. The maximum robot RMS error in azimuthal angle (ϕ) is 6.5×10^{-2} rad (or 0.71 cm in geodesic), which shows that the robot motion exhibits larger deviations in tracking the interactive tasks with respect to the planned motion, which was expected. Nevertheless, robot execution errors remain within acceptable margins, as imposed by the hardware (kinematic and control) limitations of the available robotic arm, while the performance is in any case consistent (with no significant differences) in both stationary and deforming conditions, when using the proposed IMP approach, which constitutes an important conclusive outcome of this experimental study.

Analysing comparatively the motion planning accuracy of our previous work [80] in a scene without occlusion (indicated as DVP in Table I), it is evident from the different error metrics shown in the Table that the visual noise inserted by the camera reduces significantly the motion planning accuracy. More specifically, for the radial distance (r), the largest errors are observed in the deforming scenarios (with the maximum RMSE value of 1.3 cm observed in the high frequency circular scenario). For the polar (θ) and azimuthal (ϕ) angles, which are characteristic values for the geodesic distances, the maximum errors are 8.1×10^{-2} rad (geodesic 0.89 cm) and 11×10^{-2} rad (geodesic 1.16 cm), respectively. The position planning inaccuracies occur due to depth estimation noise especially at the edge of the object despite the accurate results of the segmentation algorithm. In terms of orientation (α), the maximum error is 9.7×10^{-2} rad ($5.53°$). The overall planning performance of a direct vision-based algorithm shows that it is suitable for applications in which the visual occlusion of the surface is low and the accuracy requirements reduced. However, these planning errors are forming a baseline regarding the effect of the experimental set-up (camera noise, camera to robot registration errors) on the planning accuracy of the IMP approach as well. In case of larger visual occlusion of the object's surface by the motion of the robot (denoted as DVPO in Table I), the planning accuracy is further reduced.

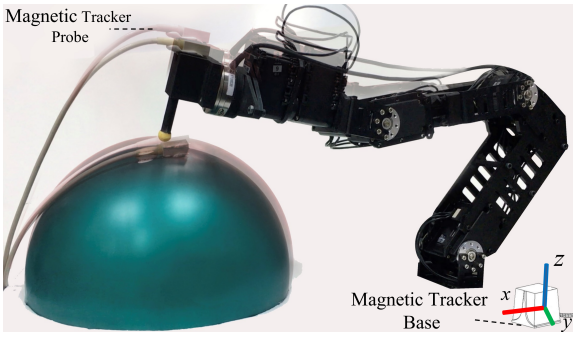


Fig. 8: Tracking of a physical point on the surface of a deformable object. The tracking accuracy is measured with millimeter accuracy using an Ascension TrakStar ATC3DGT magnetic tracker system with two probes, one attached on the robot end-effector and another one placed on the object surface.

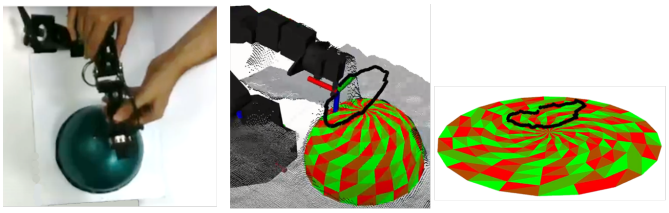
TABLE II: Physical point tracking errors presented along the axes of the Physical Space. The tracking error is calculated as the relative position between the magnetic tracker's probe attached on the deformable model and the tracker's probe attached on the robot's end-effector, subtracting the initial offset between the two tracking elements at each measurement.

Tracker experiment	**x** (cm)	**y** (cm)	**z** (cm)
f = 0.25 Hz	0.0 ± 0.1	0.0 ± 0.1	0.2 ± 0.3
f = 0.5 Hz	0.0 ± 0.1	0.0 ± 0.1	0.1 ± 0.4

Data presented as mean and standard deviation of the error values.

In this case, the maximum RMS error values obtained for the radial distance (r), the polar angle (θ) and the azimuthal angle (ϕ), are: 2.8 cm, 13×10^{-2} rad (geodesic 1.46 cm) and 74×10^{-2} rad (geodesic 8.16 cm), respectively. These high error values are in fact at least an order of magnitude larger than the ones obtained with the IMP approach. This result clearly reveals that a direct vision-based algorithm (based on local processing of raw depth data) indeed fails to properly plan the required interactive tasks, as opposed to the interactive mesh-based motion planning framework proposed in this paper, which is able to accurately plan the required trajectories on a deformable surface even in the presence of large and persistent occlusions (in our case, induced by the robot continuously moving in front of the object surface in the scene).

In order to further evaluate the motion planning accuracy of our system, we measure quantitatively the ability of the proposed framework to track a physical point on the surface of a deformable object, during the same periodic deformation scenario. In this experiment, we follow a slightly different planning strategy, in which we make use of transformation ζ from the SCS to the Physical Space, in order to preserve the geodesic distances and track the same physical point on the object's surface. Error measurements in this experiment are realized with the use of a magnetic tracking device (an Ascension TrakStar ATC3DGT magnetic tracker), which can provide position measurements with an accuracy of one millimeter. In particular, one of the magnetic tracker probes is placed on the object surface and the other probe is attached on the robot's end-effector, as depicted in Fig. 8. Table II presents



(a) Demonstration **(b)** Physical Space **(c)** Static Canonical Space

Fig. 9: The interactive trajectory demonstration and transformation procedure. **(a)** The user is shown to move the robot over the surface of a hemispherical deformable model, in a stationary instance with radius $R = 11$cm, from the system's camera perspective. **(b)** Visualization of the interactive trajectory with black line as recorded in the Physical Space. At each time step the point of the trajectory is associated with the closest underlying triangle and is transformed to the Static Canonical Space using the proposed framework. **(c)** Visualization of the interactive trajectory with black line as transformed and recorded in the Static Canonical Space.

the results obtained regarding the relative position between the two probes along each axis of the tracker reference frame, for two object deformation frequencies.

Before discussing these results, it should be noted that the relative placement of the probes (Fig. 8), upon initialisation of each experiment, adds an inevitable bias to the measurements. This initial offset has been deduced from the results presented in Table II, in order to characterize the tracking performance as measured by the mean and standard deviation values of the relative displacement error between the robot end-effector and the desired target frame on the deforming surface. Larger tracking deviation values along the z-axis compared to the x and y axes, are observed due to the placement of the probe near the pole of the hemispherical object, whose major motion direction, caused by the object's deformation, is along the z-axis. The mean tracking error does not exceed the value of 0.2 cm along z, with standard deviation errors measured at 0.4 cm, respectively, confirming the sub-centimeter accuracy performance of our approach. Similarly to the performance assessment results presented above, the robot motion is affected by its kinematic constraints and stiffness controller gains.

*2) **Interactive Motion Demonstration**:* In the previous experiments, the accuracy of the proposed framework was evaluated by planning several paths in SCS and projecting the trajectories to the Physical Space, over the surface of the target object, during real-time execution of the motion. In this section, we aim at evaluating the framework in a different context, that of motion programming by human demonstration. Our goal here is to showcase that the proposed framework can be effectively applied in such a context and is particularly suitable for planning (by means of human demonstration) robot manipulation tasks that involve interaction with the surface of a (potentially deformable) object.

For this purpose, we recall that, as shown in Section IV, each transformation (between SCS, DCS, and Physical Space) is invertible; hence, using the proposed approach, interactive tasks performed on the surface of a stationary object can be transformed back to SCS during the demonstration process, as depicted in Fig. 9. To handle interactive trajectories which lie over or under the surface (that is, involving or not physical contact with the object during task execution), an algorithm is implemented which projects the trajectory points at each time

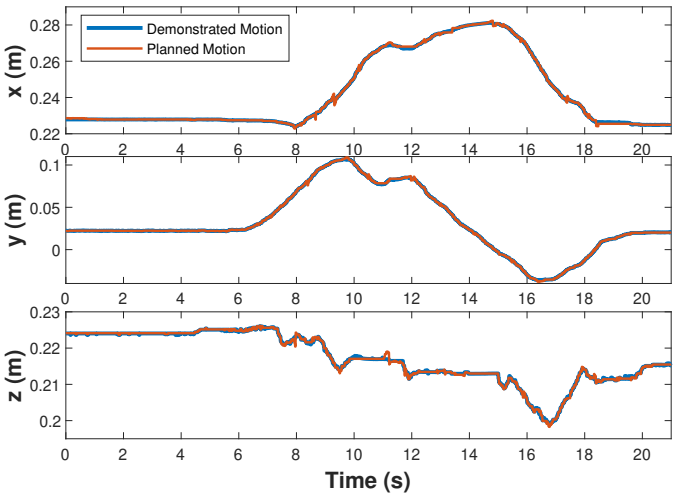


Fig. 10: Evolution over time of the Demonstrated Motion (blue) in the 3D Physical Space, compared to the reproduced motion over a stationary hemispherical model (with the same radius $R = 11$cm for both motions).

TABLE III: Motion reproduction errors of the demonstrated interactive trajectory, expressed in spherical coordinates (r, θ, ϕ), for the whole duration of motion (RMSE $\pm$ standard deviation of the error values). The motion planning errors of the proposed approach (denoted as IMP) are distinguished from the robot execution errors (evaluated on the trajectory actually executed by the robotic manipulator, denoted as Robot).

Demonstration Trajectory		**Stationary**	**Deforming**	
		R = 10cm	*f = 0,25Hz*	*f = 0,5Hz*
r	IMP	0.2 ($\pm$0.0)	0.3 ($\pm$0.0)	0.3 ($\pm$0.1)
(cm)	Robot	0.9 ($\pm$0.3)	1.1 ($\pm$1.1)	1.0 ($\pm$0.9)
$\boldsymbol{\theta}$	IMP	1.6 ($\pm$0.8)	1.6 ($\pm$0.9)	1.6 ($\pm$0.9)
(rad $\times 10^{-2}$)	Robot	5.1 ($\pm$3.7)	8.6 ($\pm$6.0)	8.2 ($\pm$7.3)
$\boldsymbol{\phi}$	IMP	1.3 ($\pm$1.3)	1.5 ($\pm$1.7)	1.5 ($\pm$1.5)
(rad $\times 10^{-2}$)	Robot	7.6 ($\pm$5.0)	14.8 ($\pm$9.4)	14.4 ($\pm$10.4)

Data presented as RMSE ($\pm$ standard deviation of the error value).

frame on the surface of the object, using shaded normals combined with the notion of active triangle described in Section IV. The continuity of the demonstration trajectories ensures that, at each time frame, the motion evolves on the same active triangle or is transferred on one of its neighbors, thus reducing the search area during execution of the algorithm.

Following this process, the demonstrated task is thus recorded (and eventually encoded, for generalisation purposes) in SCS. By applying the proposed IMP approach, the trajectory can then be properly reproduced in real time and interactively at any deformation state of the target object, respecting both the time and spatial constraints (vertical and geodesic distances) of the demonstrated task. In this experiment particularly, a complex motion is demonstrated over the surface of the hemispherical model, with radius $R = 11$ cm. In order to evaluate the accuracy of the trajectory recorded in SCS, the motion was reproduced from SCS and transformed back to Physical Space. The reproduced motion was compared to the demonstrated one, as shown in Fig. 10, illustrating that the demonstrated path is actually reproduced quite accurately back on the Physical Space, using the proposed IMP framework. Specifically, the maximum planning errors, obtained for the whole duration of motion, were measured to be: 0.37 cm along

x, 0.42 cm along y and 0.26 cm along z direction. Some motion planning disturbances, which can be observed as sparse abrupt changes (small spikes) in Fig. 10 are mostly due to the presence of noise and inconsistencies at the visual perception layer, which may affect the mesh structure tracking in the Physical space and consequently the mesh mapping in the Canonical space, where the motion is generated. Nevertheless, all such disturbance instances did not excite any observed unwanted dynamics in the robot and did not lead to any excessive variations in the torque signals as set by the controller (also validated by strict safety torque limits applied during the experiments conducted in this work). Thus, no further filtering (at the control rate) was applied in the motion planning signals, since also our objective for the purposes of this work was to validate experimentally the performance of the proposed IMP framework as being directly applied for providing updated estimates of the reference pose.

Using spherical coordinates, as described in the experiments of the previous Section, we further measured quantitatively the motion reproduction accuracy in a stationary scenario, for a hemispherical model of radius $R = 10$ cm, as well as in a deforming scenario with two different deformation frequencies ($f = 0.25$ Hz and $f = 0.5$ Hz). The obtained results are shown in Table III, where the maximum values observed for the motion planning RMS errors are: 0.3 cm error for the radius r, 1.6×10^{-2} rad and 1.5×10^{-2} rad (0.16 cm geodesic) for the polar (θ) and azimuthal (ϕ) angles, respectively (showing in fact similar sub-centimeter accuracy performance as in the simpler trajectories of the previous experiments). The robot executes the reproduced trajectory relative to the target object surface, with an accuracy which is comparable to the performance achieved in the experiments of the previous section, as revealed from the values of Table III. Consistent performance is also observed both for the stationary and the deforming scenarios, highlighting again the capacity of the proposed approach to compensate for the dynamic motions induced by the deformation of the target object surface. Furthermore, the sub-centimeter standard deviation values observed for the motion planning errors along the radial direction (IMP error values for r), which are achieved even in the deforming scenario, show a reliable motion reproduction precision throughout the duration of the motion, demonstrating again an excellent interactive and real-time motion compensation capacity. Therefore, it can be concluded that, by applying the proposed IMP framework, a demonstrated robot manipulation task (consisting of a motion relative to a surface) can be effectively recorded and properly reproduced, interactively coping with elastic deformations of the target object.

*3) **Contact Force Regulation**:* The previous series of experiments allowed for a thorough evaluation of motion planning accuracy for non-contact interactive tasks. The goal of this last experiment, is to evaluate the ability of the proposed framework to regulate the contact forces during the execution of tasks that involve physical contact with the deformable object. For this purpose, an ATI Mini-40 F/T sensor is attached to the robot's end-effector, measuring the contact forces along the direction vertical to the object's surface, as depicted in Fig. 11. As a reminder, planning of a contact task in our approach, as has been explained in Section IV, is implemented indirectly through an active stiffness controller, by setting the normal offset d to negative values (thus instructing in fact the robot to move below the object's surface, as shown in Fig. 11).

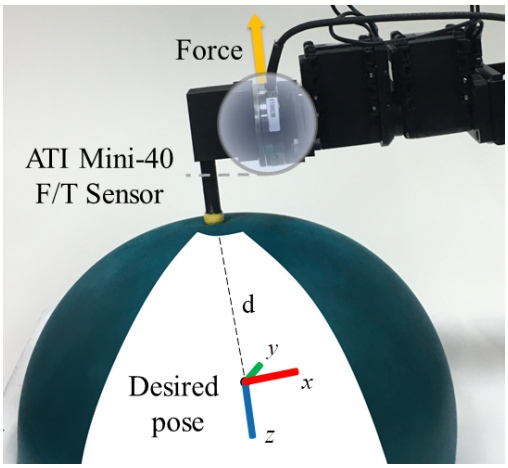
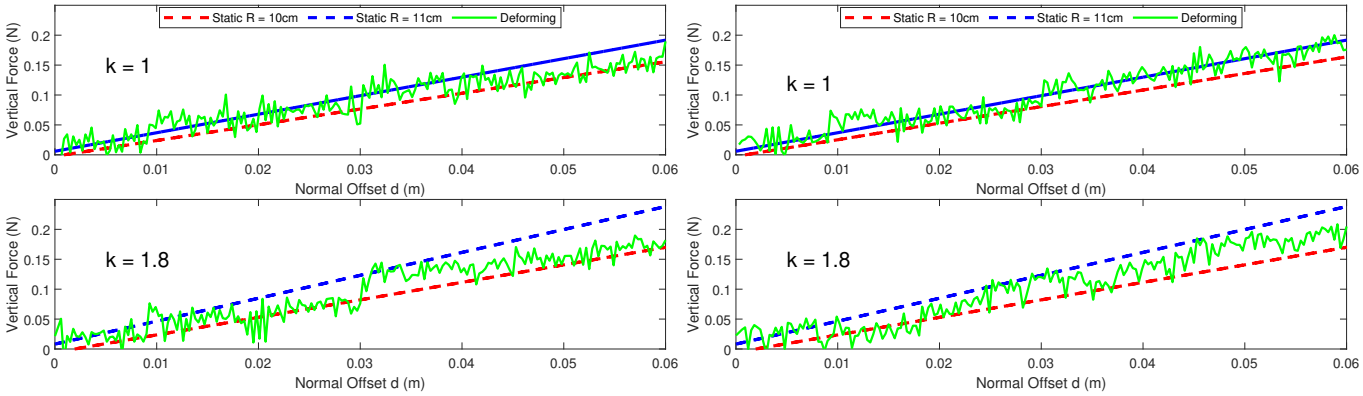


Fig. 11: Measurement of contact forces along the vertical direction (yellow arrow) to the object's surface with an ATI Mini-40 F/T sensor. The planning of a contact task is implemented in our approach by setting the normal offset d to negative values, instructing the robot to move below the object's surface.

More specifically, in this experiment, a contact task is specified such that the reference orientation remains vertical to the surface, while the normal offset linearly decreases until a minimum value of $d = -6$ cm (meaning that the tip of the robot, starting from initial contact, will start pushing on the deformable surface with a reference force that is linearly increasing). However, as mentioned in Section V-B describing the experimental setup, the deformation of the object is controlled by periodically inflating and deflating a balloon. Hence, the physical properties of the deformable model will change non-linearly for different spherical radius values, since the amount of air in the balloon is different. Therefore, in order to explore the physical properties of the object, we start by applying the previously specified contact task in a stationary scenario for two different spherical radius values, $R_{\max} = 11$ cm and $R_{\min} = 10$ cm, as shown with dashed blue and red lines in Fig. 12. Several gain values for the active stiffness controller were tested and a representative behaviour is shown in the graphs for relative gain value $k = 1$ (Fig. 12 top) and $k = 1.8$ (Fig. 12 bottom).

Apart from the controller stiffness, the relation of the normal offset d to the measured force values needs to be evaluated taking also into account the stiffness of the deformable object. In fact, what is actually observed in the experimental results is a series of stiffnesses, comprising both the object stiffness and the controller stiffness. This object-controller coupled stiffness can indeed be assessed in the experimental data by comparing the relation between the measured force and actual normal offset values for each stationary scenario (with the same spherical radius R). Such a detailed analysis shows that, for instance, in the static case with $R = 10$ cm, a relative increase of approximately 25% in the coupled stiffness should be anticipated and has indeed been observed in the measured data, despite the 80% increase in the controller stiffness (i.e. corresponding to the change in the controller gain from $k = 1$



(a) Stationary vs 0.25Hz Deforming Scenario Contact Force

(b) Stationary vs 0.5Hz Deforming Scenario Contact Force

Fig. 12: Measurement of contact forces along the vertical direction to the object's surface for a range of normal offset d. A contact task is planned such that the reference orientation is vertical to the surface and the normal offset is linearly decreased until the 6 cm value, for the stationary scenario for hemispherical radius $R_{\max} = 11$ cm and $R_{\min} = 10$ cm (dashed blue and red lines lines were fitted respectively for data presentation clarity) and for the deforming scenario (green solid line) with deformation rate of 0.25 Hz (left) & 0.5 Hz (right). Several gain values for the stiffness controller were tested and a representative behaviour is shown for relative gain value $k = 1$ (top) and $k = 1.8$ (bottom).

to $k = 1.8$ between the two experimental conditions depicted in the results). These changes in the actual stiffness values reveal the expected desired limits in the exerted forces subject to a linear increase of the normal offset, corresponding to the two static cases (for $R_{\min} = 10$ cm and $R_{\max} = 11$ cm). To highlight these limits, straight lines were fitted for data presentation clarity in Fig. 12, to illustrate the stiffness slope margins corresponding to the limit spherical radius values.

The specified contact task was then implemented with the same controller gains, in a scenario involving a periodic deformation of the object with a spherical radius varying between $R_{\max}$ and $R_{\min}$. Two different deformation rates were tested, 0.25 Hz and 0.5 Hz, and the obtained results are depicted with the green line plots in Fig. 12a and Fig. 12b, respectively. It is evident that the proposed interaction framework is able to compensate for elastic deformations of the object and maintain continuous contact with the object surface throughout the task, without applying excessive forces along the vertical direction. This behaviour is apparently achieved by means of an accurate and reactive adaptation to the changes occurring due to object deformation. It can thus be concluded that, by tracking the surface deformation and interactively adapting the reference pose (through the offset values) within the proposed IMP framework, the contact forces can be regulated and remain within the specified range of values. Therefore, this experiment should be considered as a *proof-of-concept* showing that the proposed framework can directly generalise to contact tasks and is able to compensate for the object deformation, maintaining bounded forces along the vertical direction to a deforming surface.

VI. Conclusion & Future Work

In this paper, we presented an efficient integrated motion planning framework to effectively and accurately control a robotic manipulator when executing interactive tasks on the surface of a deformable object. The proposed interactive motion planning (IMP) framework integrates 3D visual object segmentation and on-line deformation modeling with real-time motion planning and interaction control. The core of the system is based on FEM deformation tracking and efficient local mesh parameterization techniques, and uses barycentric coordinates defined on the mesh triangles to establish bijective transformations between the deformable part of an object surface and its planar (static and dynamic) parameterized versions. By combining these spatial transformations with the preprocessing (visual perception and deformation modeling) steps, we can reactively plan interactive trajectories with high accuracy, even under large and persistent visual occlusions (such as those caused by the presence of the robot manipulator in the visual scene). In the conducted experimental evaluation study with a periodically deforming hemispherical model, sub-centimeter accuracy was achieved for motion planning at all conditions with three different trajectories tested, clearly showing superior performance as compared to our previous direct vision-based approach. The accuracy in tracking a physical point on the surface of a deforming object was also assessed objectively with a motion tracking device, confirming high precision performance. It is obvious, however, that the *planning accuracy* is highly dependent on the performance of the algorithms used as preprocessing steps. Hence, the choice and development of visual object segmentation, physics based deformation tracking and mesh parameterization algorithms is important and application dependent. The *robot accuracy* is measured, separately from the planning accuracy, through forward kinematics or by means of external sensors (magnetic tracker), to avoid the influence of the hardware limitations associated to the available robot (such as, due to mechanics and control bandwidth, as well as caused by robot calibration during assembly), also shown in the video attachment to this paper, which aims at demonstrating the overall system performance showcasing all the experimental configurations and their objectives as described in the paper.

Furthermore, it was shown experimentally that the proposed IMP approach can also be effectively applied in the context of motion programming by human demonstration. It was showcased that demonstration of an interactive motion task can be performed by a human expert in an undeformed state of the object, and this interactive motion relative to the target

object surface can be properly projected and recorded in the Static Canonical Space. The demonstrated actions can then be reproduced precisely even when the object undergoes deformation, exemplifying how a robotic system equipped with the proposed IMP approach can work collaboratively with the user and eventually enhance his/her ability to perform tasks involving interaction with the surface of a deformable object. The performance of the proposed framework was also further evaluated with the execution of interactive tasks in continuous physical contact with the deformable surface of an object (by measuring the exerted contact forces with a f/t sensor mounted on the robot). It was shown experimentally that the integrated framework manages to compensate for homogeneous elastic deformation of the object and to maintain continuous contact with the target surface, while exerting contact forces that remain within a specified range of values. It can thus be concluded that, by employing the proposed integrated IMP framework, it is possible to accurately control interactive path tracking tasks on the surface of a dynamically deforming object, while also effectively regulating contact forces.

This research work has multiple extensions regarding both planning and control as well as learning of interactive tasks. For example, the execution of interactive trajectories in continuous contact with a deformable object requires the exploration of impedance robot control schemes, integrating force measurements into the control loop with deformation modelling, to make the physical interaction as smooth as possible taking into account all the imposed constraints. Such a task would also involve more complex contact wrench regulation in multiple directions, combining both the manipulator and the object deformation dynamics. Additionally, robotic interaction with the surface of more complex shaped objects should be examined, along with the ability of the framework to control the object's deformation or to estimate the object's physical parameters by planning appropriately the robot's motion and actions. The ability of the framework to support efficient learning and generalisation of such complex tasks will also be examined in future work, particularly concerning human-robot collaboration in tasks that involve interaction with dynamically deforming objects. Finally, leveraging the computational efficiency of spatial mesh transformations, this framework can be easily extended to planning the motion of multiple robots or of a robotic hand with multiple fingers for grasping actions.

ACKNOWLEDGMENT

The authors would like to thank Antoine Petit and Matteo Mastellaro for their valuable support during the experimental evaluation.

REFERENCES

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019.

[2] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, and J. L. Wyatt, "Modeling of deformable objects for robotic manipulation: A tutorial and review," *Frontiers in Robotics and AI*, vol. 7, p. 82, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2020.00082

[3] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," in *Computer graphics forum*, vol. 25, no. 4. Wiley Online Library, 2006, pp. 809–836.

[4] P. Moore and D. Molloy, "A survey of computer-based deformable models," in *International Machine Vision and Image Processing Conference (IMVIP 2007)*. IEEE, 2007, pp. 55–66.

[5] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, Dec 2010.

[6] R. Szeliski and D. Tonnesen, *Surface modeling with oriented particle systems*. ACM, 1992, vol. 26, no. 2.

[7] F. F. Khalil and P. Payeur, "Dexterous robotic manipulation of deformable objects with multi-sensory feedback-a review," in *Robot Manipulators Trends and Development*. IntechOpen, 2010.

[8] F. Nadon, A. J. Valencia, and P. Payeur, "Multi-modal sensing and robotic manipulation of non-rigid objects: A survey," *Robotics*, vol. 7, p. 74, 2018.

[9] 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," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[10] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking." in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, vol. 11, no. 2011, 2011, pp. 127–136.

[11] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 343–352.

[12] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3d reconstruction with loop closure," in *European Conference on Computer Vision*. Springer, 2016, pp. 500–516.

[13] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt *et al.*, "Real-time non-rigid reconstruction using an RGB-D camera," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 4, p. 156, 2014.

[14] R. Alp Guler and I. Kokkinos, "Holopose: Holistic 3d human reconstruction in-the-wild," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] A. Tsoli and A. A. Argyros, "Tracking deformable surfaces that undergo topological changes using an RGB-D camera," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 333–341.

[16] M. Famouri, A. Bartoli, and Z. Azimifar, "Fast shape-from-template using local features," *Machine Vision and Applications*, vol. 29, no. 1, pp. 73–93, 2018.

[17] B. Willimon, I. Walker, and S. Birchfield, "3d non-rigid deformable surface estimation without feature correspondence," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 646–651.

[18] A. Jordt and R. Koch, "Direct model-based tracking of 3d object deformations in depth and color video," *International journal of computer vision*, vol. 102, no. 1-3, pp. 239–255, 2013.

[19] F. Hui, P. Payeur, and A.-M. Cretu, "Visual tracking of deformation and classification of non-rigid objects with robot hand probing," *Robotics*, vol. 6, no. 1, p. 5, 2017.

[20] C. M. Mateo, P. Gil, D. Mira, and F. Torres, "Analysis of shapes to measure surfaces: An approach for detection of deformations," in *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 2. IEEE, 2015, pp. 60–65.

[21] J. Tian and Y.-B. Jia, "Modeling deformations of general parametric shells grasped by a robot hand," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 837–852, 2010.

[22] B. Tawbe and A.-M. Cretu, "Acquisition and neural network prediction of 3d deformable object shape using a kinect and a force-torque sensor," *Sensors*, vol. 17, no. 5, p. 1083, 2017.

[23] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1130–1137.

[24] A. Petit, V. Lippiello, and B. Siciliano, "Real-time tracking of 3d elastic objects with an RGB-D sensor," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 3914–3921.

[25] A. Petit, V. Lippiello, G. A. Fontanelli, and B. Siciliano, "Tracking elastic deformable objects with an RGB-D sensor for a pizza chef robot," *Robotics and Autonomous Systems*, vol. 88, pp. 187–201, 2017.

[26] H. G. Tanner, "Mobile manipulation of flexible objects under deformation constraints," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 179–184, 2006.

[27] Y. Li, Y. Wang, M. Case, S.-F. Chang, and P. K. Allen, "Real-time pose estimation of deformable objects using a volumetric approach," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1046–1052.

[28] Y. Li, D. Xu, Y. Yue, Y. Wang, S.-F. Chang, E. Grinspun, and P. K. Allen, "Regrasping and unfolding of garments using predictive thin shell modeling," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1382–1388.

[29] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 6000–6006.

[30] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[31] A. Doumanoglou, A. Kargakos, T.-K. Kim, and S. Malassiotis, "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 987–993.

[32] D. Triantafyllou, I. Mariolis, A. Kargakos, S. Malassiotis, and N. Aspragathos, "A geometric approach to robotic unfolding of garments," *Robotics and Autonomous Systems*, vol. 75, pp. 233–243, 2016.

[33] A. Namiki and S. Yokosawa, "Robotic origami folding with dynamic motion primitives," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5623–5628.

[34] X. Li, X. Su, and Y. Liu, "Cooperative robotic soldering of flexible pcbs," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1651–1656.

[35] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez, "Fem-based deformation control for dexterous manipulation of 3d soft objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4007–4013.

[36] J. Das and N. Sarkar, "Planning and control of an internal point of a deformable object," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2877–2882.

[37] D. Navarro-Alarcón, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1457–1468, 2013.

[38] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4525–4532.

[39] 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," *Robotics and Autonomous Systems*, vol. 95, pp. 196–206, 2017.

[40] H. Lin, F. Guo, F. Wang, and Y.-B. Jia, "Picking up a soft 3d object by "feeling" the grip," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1361–1384, 2015.

[41] A. R. Fugl, A. Jordt, H. G. Petersen, M. Willatzen, and R. Koch, "Simultaneous estimation of material properties and pose for deformable objects from depth and color images," in *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*. Springer, 2012, pp. 165–174.

[42] B. Frank, C. Stachniss, R. Schmedding, M. Teschner, and W. Burgard, "Learning object deformation models for robot motion planning," *Robotics and Autonomous Systems*, vol. 62, no. 8, pp. 1153–1174, 2014.

[43] 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 *ICINCO 2017-14th International Conference on Informatics in Control, Automation and Robotics*, vol. 2, no. 978-989-758-263-9. ScitePress; Springer, 2017, pp. 24–33.

[44] S. Caccamo, P. Güler, H. Kjellström, and D. Kragic, "Active perception and modeling of deformable surfaces using gaussian processes and position-based dynamics," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 530–537.

[45] P. Güler, A. Pieropan, M. Ishikawa, and D. Kragic, "Estimating deformability of objects using meshless shape matching," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5941–5948.

[46] L. Maier-Hein, P. Mountney, A. Bartoli, H. Elhawary, D. Elson, A. Groch, A. Kolb, M. Rodrigues, J. Sorger, S. Speidel, and D. Stoyanov, "Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery," *Medical Image Analysis*, vol. 17, no. 8, pp. 974 – 996, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841513000480

[47] T. R. dos Santos, A. Seitel, T. Kilgus, S. Suwelack, A.-L. Wekerle, H. Kenngott, S. Speidel, H.-P. Schlemmer, H.-P. Meinzer, T. Heimann *et al.*, "Pose-independent surface matching for intra-operative soft-tissue marker-less registration," *Medical Image Analysis*, vol. 18, no. 7, pp. 1101–1114, 2014.

[48] 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 *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 199–208.

[49] N. Haouchine, J. Dequidt, M.-O. Berger, and S. Cotin, "Monocular 3d reconstruction and augmentation of elastic surfaces with self-occlusion handling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 12, pp. 1363–1376, 2015.

[50] A. Malti, A. Bartoli, and T. Collins, "Template-based conformal shape-from-motion from registered laparoscopic images." in *MIUA*, vol. 1, no. 2, 2011, p. 6.

[51] M. N. Cheema, A. Nazir, B. Sheng, P. Li, J. Qin, J. Kim, and D. Feng, "Image-aligned dynamic liver reconstruction using intra-operative field of views for minimal invasive surgery," *IEEE Transactions on Biomedical Engineering*, 2018.

[52] I. Leizea, H. Álvarez, I. Aguinaga, and D. Borro, "Real-time deformation, registration and tracking of solids based on physical simulation," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2014, pp. 165–170.

[53] R. E. Goldman, A. Bajo, and N. Simaan, "Algorithms for autonomous exploration and estimation in compliant environments," *Robotica*, vol. 31, no. 1, pp. 71–87, 2013.

[54] S. Patil and R. Alterovitz, "Toward automated tissue retraction in robot-assisted surgery," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2088–2094.

[55] M. Dominici, P. Poignet, R. Cortesão, E. Dombre, and O. Tempier, "Compensation for 3d physiological motion in robotic-assisted surgery using a predictive force controller. experimental results," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 2634–2639.

[56] L. Cheng, J. Fong, and M. Tavakoli, "Semi-autonomous surgical robot control for beating-heart surgery," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1774–1781.

[57] A. Ruszkowski, C. Schneider, O. Mohareri, and S. Salcudean, "Bimanual teleoperation with heart motion compensation on the da vinci® research kit: Implementation and preliminary experiments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4101–4108.

[58] S. Leonard, K. L. Wu, Y. Kim, A. Krieger, and P. C. Kim, "Smart tissue anastomosis robot (STAR): A vision-guided robotics system for laparoscopic suturing," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1305–1317, 2014.

[59] J. D. Langsfeld, A. M. Kabir, K. N. Kaipa, and S. K. Gupta, "Integration of planning and deformation model estimation for robotic cleaning of elastically deformable objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 352–359, 2017.

[60] A. Zlatintsi, A. Dometios, N. Kardaris, I. Rodomagoulakis, P. Koutras, X. Papageorgiou, P. Maragos, C. S. Tzafestas, P. Vartholomeos, K. Hauer *et al.*, "I-support: A robotic platform of an assistive bathing robot for the elderly population," *Robotics and Autonomous Systems*, vol. 126, p. 103451, 2020.

[61] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Science Robotics*, vol. 1, no. 1, p. eaah3690, 2016.

[62] S. Lee, "Development of a new variable remote center compliance (vrcc) with modified elastomer shear pad (esp) for robot assembly," *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 2, pp. 193–197, 2005.

[63] L. Villani and J. De Schutter, "Force control," *Springer handbook of robotics*, pp. 161–185, 2008.

[64] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[65] R. Richa, A. P. Bó, and P. Poignet, "Towards robust 3d visual tracking for motion compensation in beating heart surgery," *Medical Image Analysis*, vol. 15, no. 3, pp. 302–315, 2011.

[66] D. Stoyanov and G.-Z. Yang, "Soft tissue deformation tracking for robotic assisted minimally invasive surgery," in *2009 annual interna-*

tional conference of the IEEE engineering in medicine and biology society. IEEE, 2009, pp. 254–257.
[67] B. Zhao, J. Feng, X. Wu, and S. Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," *International Journal of Automation and Computing*, vol. 14, no. 2, pp. 119–135, 2017.
[68] N. M. Zaitoun and M. J. Aqel, "Survey on image segmentation techniques," *Procedia Computer Science*, vol. 65, pp. 797–806, 2015.
[69] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd, "Coarse-to-fine low-rank structure-from-motion," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.
[70] A. Weiss, D. Hirshberg, and M. J. Black, "Home 3d body scans from noisy image and range data," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1951–1958.
[71] C. Elbrechter, R. Haschke, and H. Ritter, "Bi-manual robotic paper manipulation based on real-time marker tracking and physical modelling," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1427–1432.
[72] M. S. Floater and K. Hormann, "Surface parameterization: a tutorial and survey," in *Advances in multiresolution for geometric modelling*. Springer, 2005, pp. 157–186.
[73] C. Wang, Z. Liu, and L. Liu, "As-rigid-as-possible spherical parametrization," *Graphical models*, vol. 76, no. 5, pp. 457–467, 2014.
[74] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization," in *Computer Graphics Forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1495–1504.
[75] R. Schmidt, C. Grimm, and B. Wyvill, "Interactive decal compositing with discrete exponential maps," in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 605–613.
[76] W. T. Tutte, "How to draw a graph," *Proceedings of the London Mathematical Society*, vol. 3, no. 1, pp. 743–767, 1963.
[77] P. Kyriakis and G. Moustris, "Terrain following for fixed-wing unmanned aerial vehicles using feedback equivalence," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 150–155, 2019.
[78] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3d path planning algorithms," *Journal of Control Science and Engineering*, vol. 2016, 2016.
[79] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.
[80] A. C. Dometios, X. S. Papageorgiou, A. Arvanitakis, C. S. Tzafestas, and P. Maragos, "Real-time end-effector motion behavior planning approach using on-line point-cloud data towards a user adaptive assistive bath robot," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5031–5036.

Athanasios C. Dometios received the PhD degree in Robotics from the Intelligent Robotics & Automation Lab (IRAL - http://robotics.ntua.gr) of the National University of Athens, Greece, in 2021. He received his Engineering Diploma from the School of Electrical and Computer Engineering (ECE) of the same University. As a research engineer, he has more than 5 years of R&D experience in several EU and National projects, involved in the development of robotics and autonomous systems. He completed his Diploma Thesis at the Institute of Automatic Control Engineering, Technische Universität München (TUM) under the Erasmus Program. His research interests include human-robot physical interaction, intelligent control and robot learning for advanced manipulation, reactive planning and cognitive assistive robotics.

Costas S. Tzafestas holds an Electrical and Computer Engineering (ECE) Degree from the National Technical University of Athens (NTUA) (1993), as well as a D.E.A. (1994) and Ph.D. (1998) Degrees on Robotics from the Université Pierre et Marie Curie (UPMC/Paris 6, France). In 2003 he joined NTUA School of ECE where he currently serves as an Associate Professor of Robotics (http://robotics.ntua.gr/members/ktzaf/) His main research interests include cognitive assistive robotics, human-robot interaction, haptics and telerobotics, also spanning intelligent control and robot learning with applications in advanced manipulation and mobile robotics. He has authored or co-authored more than 150 scientific publications and has participated as Project Coordinator, PI, or Scientific Manager in several European and national research projects in the above fields. He currently serves as an Editor-at-Large of the Journal of Intelligent and Robotic Systems.