Article:
Hu, J. orcid.org/0000-0001-7394-5580, Jones, D., Dogar, M. et al. (1 more author) (2023) Occlusion-Robust Autonomous Robotic Manipulation of Human Soft Tissues With 3D Surface Feedback. IEEE Transactions on Robotics, 40. pp. 624-638. ISSN 1552-3098

https://doi.org/10.1109/TRO.2023.3335693

Occlusion-Robust Autonomous Robotic Manipulation of Human Soft Tissues with 3D Surface Feedback

Junlei Hu, Dominic Jones *Member, IEEE*, Mehmet R. Dogar, Pietro Valdastri, *Fellow, IEEE*

***Abstract*—Robotic manipulation of 3D soft objects remains challenging in the industrial and medical fields. Various methods based on mechanical modelling, data-driven approaches or explicit feature tracking have been proposed. A unifying disadvantage of these methods is the high computational cost of simultaneous imaging processing, identification of mechanical properties, and motion planning, leading to a need for less computationally intensive methods. We propose a method for autonomous robotic manipulation with 3D surface feedback to solve these issues. First, we produce a deformation model of the manipulated object, which estimates the robots' movements by monitoring the displacement of surface points surrounding the manipulators. Then, we develop a 6-degree-of-freedom velocity controller to manipulate the grasped object to achieve a desired shape. We validate our approach through comparative simulations with existing methods and experiments using phantom and cadaveric soft tissues with the *da Vinci* Research Kit. The results demonstrate the robustness of the technique to occlusions and various materials. Compared to state-of-the-art linear and data-driven methods, our approach is more precise by 46.5% and 15.9% and saves 55.2% and 25.7% manipulation time, respectively.**

***Index Terms*—Shape control, dual arm manipulation, robotic manipulation of soft objects**

I. Introduction

ROBOTIC manipulation of soft, deformable objects is a common practice in industry, from fabric folding to food packaging [1], [2], [3], [4], [5], with similar problems observed in medical scenarios. Despite recent progress in the robotic manipulation of deformable linear and planar objects, such as ropes and clothing items, the shape control of 3D objects is still a challenge.

In robotic-assisted minimally invasive surgery (RAMIS) [6], soft tissues within the abdomen are manipulated with the teleoperated end-effectors of a surgical robot [7]. Recently, the research focus in this area has trended towards autonomous task execution, aiming to remove the operating surgeon from the control loop. This creates a need for robust autonomous control systems that can operate effectively in a surgical environment. Several research efforts are currently dedicated to autonomous robotic manipulation of soft objects in RAMIS, such as autonomous fat retraction [8] and intestinal anastomosis using threads manipulation [9]. However, research on increasing computer assistance when manipulating human soft tissues has not yet been explored.

Two critical complexities of robotic manipulation of 3D soft objects stem from estimating mechanical properties and representing surface shape [10]. It is uneconomical to build a patient-specific biomechanical model before surgery and impractical to identify the deformation properties of the soft tissues during manipulation tasks [11]. Regarding 3D shape representation, using fluorescent fiducial markers [9] or keypoint features [11], [12] as nodes to generate a surface may simplify the problem. However, placing such markers within the abdominal cavity is undesirable due to the risk of leaving foreign objects in the body after surgery. Current research on markerless keypoint tracking shows limited success in accuracy and framerate [13]. A more efficient way of detecting the surface shape is to use the entire feedback surface before establishing the explicit or implicit representation. In this study, we mainly focus on using the 3D point cloud of the surface as feedback to control the shape of the soft object autonomously. Based on this, we propose a controller for the robotic manipulation of 3D soft objects without prior knowledge of their underlying mechanical properties.

A. Related Work

Similar to robotic manipulation of rigid objects, the soft object manipulation controller needs to define a representative state first and then search for optimal actions based on given objective tasks, as illustrated in Fig. 1.

Unlike rigid objects, the state of a soft object cannot be described solely by orientation and translation. In systems with force sensing, deformable objects are typically modelled using physics-based approaches such as mass-spring, position-based dynamics, or continuum mechanics [10], [14], [15], and the states are designed based on the physical configurations. However, estimating mechanical properties in real-time remains challenging, leading to low-fidelity physical models and limited manipulation precision. Several recent works have achieved successful vision-guided manipulations without additional physics modelling or parameter identification [11], [16],

The work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 818045), the European Commission (EC) Horizon 2020 research and innovation programme (Grant Agreement No. 952118) and the Engineering and Physical Sciences Research Council (EPSRC) under grant numbers EP/R045291/1 and EP/V047914/1. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of ERC, EC or EPSRC.

Junlei Hu, Dominic Jones, Pietro Valdastri are with the STORM Lab, Institute of Autonomous Systems and Sensing (IRASS), School of Electronic and Electrical Engineering, University of Leeds, Leeds, UK. Email:{eljh, d.p.jones, p.valdastri}@leeds.ac.uk

Mehmet R. Dogar is with School of Computing, University of Leeds, UK. Email: m.r.dogar@leeds.ac.uk

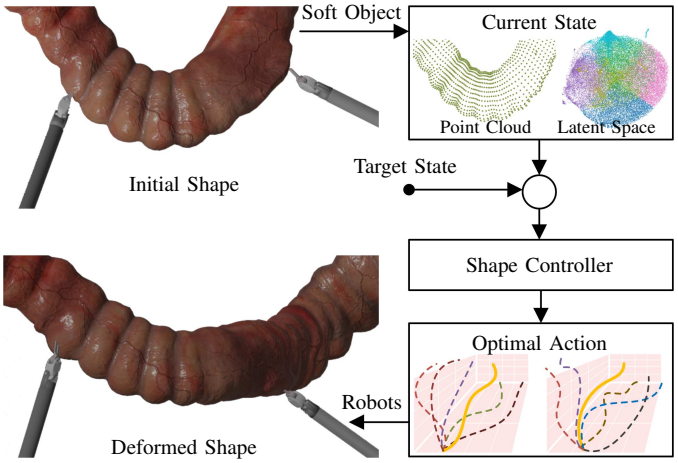


Fig. 1. An illustration of the problem of robotic manipulation of soft tissues in RAMIS. A multi-arm robot system grasps the soft tissue to deform its shape. The system state can be represented with point clouds or latent spaces.

[17], demonstrating the potential of this approach. Therefore, this study solely relies on vision information as feedback.

In vision-based manipulation, the representative state is commonly simplified as a combination of position and shape terms [11], [12], [18] with explicit features such as fiducial markers. Simplifying the problem through explicit feature definition requires increasing the computational load to track specific features. This reduces the data in the feedback loop, providing higher efficiency at the cost of lower precision in shape matching [18]. More recently, the high-dimensional image space and low-dimensional latent space have been used to represent the surface state. Working in the high dimensional image space, visual-dynamics-based states for robotic motion planning are trained on unlabelled video data at pixel level with accompanying depth maps (RGBD images) [16], producing dynamic models directly from the input images. Due to the high dimensionality of the space, the models require extremely large amounts of training data and have a high computational cost. To increase efficiency, at the trade-off of lower precision, the problem's dimensionality can be reduced by identifying underlying features in the low-dimensional latent space of the RGBD images. Typical features used as states include histogram-of-oriented-wrinkles features [1], [19], Fourier coefficients of 2D contours [20], latent manifolds [5], combinations of low-level geometric feature extraction and high-level semantic shape analysis [17], and voxelised objects [3]. While these latent-space features provide increased efficiency, there has been limited investigation of the generalisation ability of these representations, with models requiring careful state selection in various tasks to maintain the specified precision. To address the aforementioned limitations, researchers are exploring the use of depth maps or point clouds generated by stereo cameras to track deformation in 3D visual-based robotic manipulations [16], [17], [21], [22]. However, using these 3D shapes as representative states is not well-studied yet.

Once the state representation is determined, one must find the optimal action. Two common methods are model-based and learning-based approaches. In model-based methods, local linearisation, using approximated Jacobian relations, is a popular tool for closed-loop control [11], [20], [23], [24], [25]. The deformation Jacobian matrix derived from the linearised equilibrium equation can map the state to the action. Utilising linear control laws can be efficient for the global stabilisation of nonlinear systems if the parameters are well estimated.

Learning-based approaches for acquiring actions can be classified into reinforcement learning (RL) and imitation learning (IL). RL learns a control policy through trial-and-error interactions and reward collection, working well in some simple tasks [21], [26]. To handle complex state spaces in certain tasks, Deep Reinforcement Learning (DRL) [2] has been used by combining the nature of smooth policy updates with the capability of automatic feature extraction in deep neural networks. DRL enhances the sample efficiency and learning stability with fewer samples. On the other hand, IL leverages human-provided demonstrations of soft object manipulation instead of relying on sparse rewards or manually specified rewards. IL has proven effective in soft object manipulation with examples showing promise in manipulating planar soft objects such as clothes folding and dressing [1], [27], [28]. Learning-based methods typically require a large amount of data to operate effectively, necessitating repeated interactions with the soft object to allow the models to cover a broader range of scenarios. However, acquiring and labelling large datasets from real-robot interactions can be costly [10]. To address this issue, simulation-based training has been employed in previous studies [29], [30]. Nevertheless, the significant differences between simulated environments and real-world scenarios limit the applicability of simulation-trained models [8]. Furthermore, in RAMIS, variations in the mechanical properties and appearance of soft tissues render nonadaptive RL or IL models unsuitable for many surgical settings.

B. Contributions

In this work, we demonstrate the ability to control the shape of a soft object. The contribution of this work can be summarised as follows:

- The proposed 3D soft object robotic manipulation method uses the down-sampled grid points of the feedback surface as the state representation and establishes a direct non-linear mapping from shape difference to robot movement based on the weighted residual deformation model, yielding multiple 6-DoF Cartesian end-effector postures.
- The method can achieve robustness to occlusion by adjusting the down-sampling resolution.
- The proposed method for shape control has been validated in simulation and real-world setups, including experiments in autonomous RAMIS on cadaveric intestinal tissues.

II. Preliminaries

A. Continuum Deformation

The deformation gradient, $\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \mathbf{I}$ [1], determines how an infinitesimal material element $d\mathbf{X}$ changes

[1] $\mathbf{X}$ is the material coordinates, and $\mathbf{x}$ is the spatial coordinates

as a result of the deformation $\mathbf{u}$. In the finite element method (FEM) theory, $\mathbf{F}$ can be discretised via tetrahedralisation of the shape. For each tetrahedron, denoted by its four vertices $\{\mathbf{X}_{i_0}, \mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}\}$ in material coordinates and $\{\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \mathbf{x}_{i_3}\}$ in spatial coordinates, the deformation gradient at i-th vertex is represented by

$$\mathbf{F}_i = \hat{\mathbf{V}}_i \mathbf{V}_i^{-1} \tag{1}$$

where $\hat{\mathbf{V}}_i = [\mathbf{x}_{i_1} - \mathbf{x}_{i_0} \quad \mathbf{x}_{i_2} - \mathbf{x}_{i_0} \quad \mathbf{x}_{i_3} - \mathbf{x}_{i_0}] \in \mathbb{R}^{3\times3}$ and $\mathbf{V}_i = [\mathbf{X}_{i_1} - \mathbf{X}_{i_0} \quad \mathbf{X}_{i_2} - \mathbf{X}_{i_0} \quad \mathbf{X}_{i_3} - \mathbf{X}_{i_0}] \in \mathbb{R}^{3\times3}$. Thus, the partial differential equations (PDEs) can represent the relation between the deformation and reference shape using the definition and estimation of the deformation gradient:

$$\mathcal{L}\mathbf{u} = \mathbf{x} \tag{2}$$

where $\mathcal{L}$ is the differential operator.

Solving the PDEs with information about mechanical properties (e.g. Young's modulus, Poisson's ratio) is challenging. The weighted residual method (WRM) approximates differential equation solutions as a linear combination of shape functions with unknown coefficients. According to Eq. (2), the displacement of feature points can be employed as the boundary condition. The displacement field $\mathbf{u}$ of surface points is approximated using WRM on feature points. When dealing with a point cloud, the registration of the current and desired point clouds yields the displacement of some grid points:

$$\mathbf{u}(\mathbf{x}_i) = \mathbf{u}_i. \tag{3}$$

This study uses this idea to generate an approximation of the displacement field and subsequently calculate the movements of robotic manipulators to minimise the disparity.

B. Problem Formulation

Consider a soft object $\mathcal{O}$, which is grasped by a multi-robot system to be deformed into a desired shape $\mathcal{T}$. Let the surface $\mathcal{S} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$ be a set of 3D points in Cartesian frame captured by a stereo camera from $\mathcal{O}$. As shown in Fig. 2, the soft object is controlled by a multi-robot system. During the manipulation, the point cloud $\mathcal{S}$ of the soft object is captured by an RGBD camera. The multi-robot system includes $K (K > 1)$ robotic manipulators, whose positions and postures of the grippers in the spatial frame are the manipulation points, denoted as $\mathcal{R} = \{(\mathbf{R}_i^{\mathrm{m}}, \mathbf{p}_i^{\mathrm{m}}) : \mathbf{R}_i^{\mathrm{m}} \in SO(3), \mathbf{p}_i^{\mathrm{m}} \in \mathbb{R}^3\}_{i=1}^K$.

In this study, we aim to design a controller for shaping a soft object, making the following assumptions:

- The soft object is modelled as a continuous and isotropic material.
- The kinematics of the robots and the camera-robot and robot-robot relationships are known and calibrated. The robots are controlled in their joint spaces, with velocities and movements limited to their workspace.
- The robotic grippers have a firm grasp on the soft object, and the robots are rigid in their physical interaction.
- The captured surface is represented as a point cloud, with some points potentially missing due to poor reconstruction or occlusion.

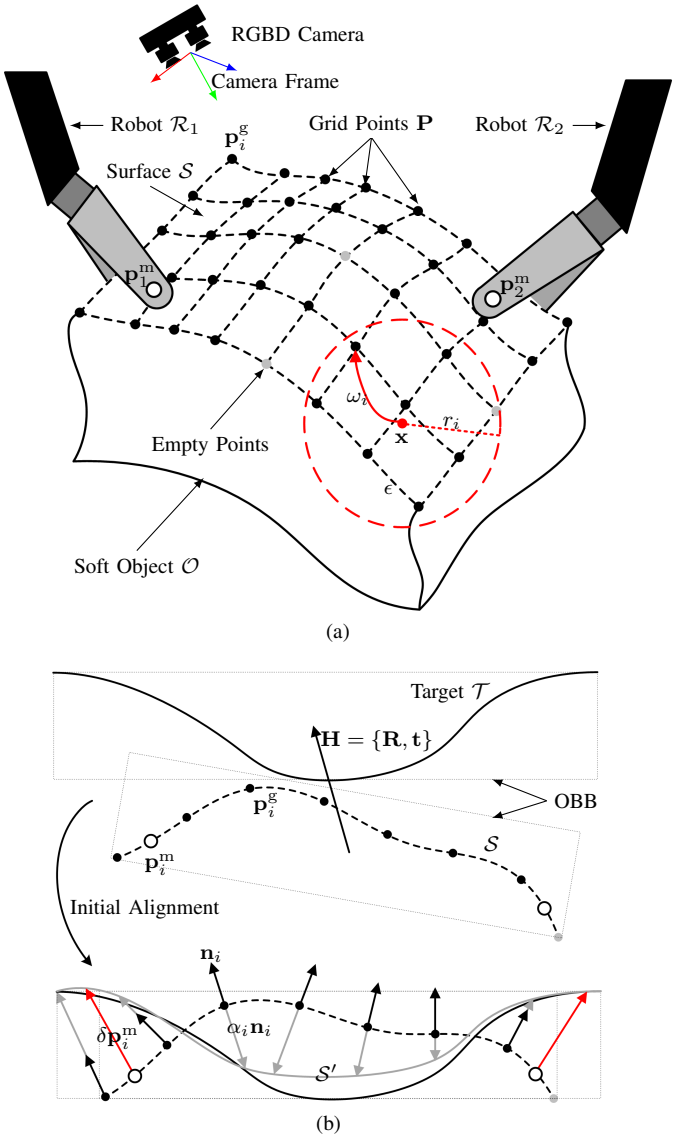


Fig. 2. (a) Illustration of 3D surface feedback in soft object manipulation. The red dashed circle defines the area around a grid point. Grey dots represent unregistered points in the point cloud. (b) Concept of displacement field computation. The current and target surfaces are shown in sectional views, aligned with the oriented bounding box (OBB). After applying the displacement field, the grey curve is the deformed surface $\mathcal{S}$. The red arrows indicate the movement of the robots.

- The target shape is planned before manipulation, based on the initial configuration of the soft object.

Remark 1: To simplify the description, all positions and rotations mentioned in the subsequent sections are referenced to the camera coordinate system, as depicted in Fig. 2(a).

Problem Statement: Given a desired (constant and physically reachable) shape in $\mathcal{T}$, develop a controller for a vision-guided multi-arm robotic system to manipulate a soft object $\mathcal{O}$ with the instant deformed surface $\mathcal{S}$, and finally minimise the difference between $\mathcal{S}$ and $\mathcal{T}$, without knowledge of the object's mechanical properties.

III. METHODS

A. Deformation Model

Eq. (2) can be numerically solved with WRM estimation. We assume the existence of a displacement field $\mathbf{u}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^3$ such that the modified surface $\mathcal{S}' = \{\mathbf{u}(\mathbf{p}) + \mathbf{p} : \forall \mathbf{p} \in \mathcal{S}\}$ is similar to the target surface $\mathcal{T}$. Here, $\mathbf{u}$ is defined discretely by assigning per-frame displacement vector $\mathbf{u}_i$ to the point i. The displacement of any position $\mathbf{x}$ in spatial coordinate is [31]

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{N} \phi_i(\mathbf{x})\, \mathbf{u}_i \tag{4}$$

where the scalar shape function $\phi_i(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}$, defined as

$$\phi_i(\mathbf{x}) = \mathbf{l}(\mathbf{x})^{\intercal} \left(\mathbf{M}(\mathbf{x})\right)^{-1} \omega_i(\mathbf{x})\, \mathbf{l}(\mathbf{x}_i). \tag{5}$$

$\phi_i(\mathbf{x})$ is constructed using moving least squares (MLS) with a complete linear basis $\mathbf{l}(\mathbf{x}) = [1 \quad \mathbf{x}^{\intercal}]^{\intercal} : \mathbb{R}^3 \to \mathbb{R}^4$, to guarantee first-order consistency. The scale factor $\omega_i(\mathbf{x}) = \max\left(0, \left(1 - \|\mathbf{x} - \mathbf{x}_i\|^2 / r_i^2\right)^3\right) : \mathbb{R}^3 \to \mathbb{R}$ is related to the distance between the points. It ensures that only points within the ball at the centre of $\mathbf{x}$ with the radius r_i will be adopted for approximation. The moment matrix $\mathbf{M}(\mathbf{x}) \in \mathbb{R}^{3\times 3}$ is

$$\mathbf{M}(\mathbf{x}) = \sum_{i=1}^{N} \omega_i \mathbf{l}(\mathbf{x}_i)\, \mathbf{l}(\mathbf{x}_i)^{\intercal}. \tag{6}$$

To avoid the singularity of $\mathbf{M}$, at least four non-planar grid points must surround the evaluation point $\mathbf{x}$. Please refer to Appendix A for the proof and solution for ill-conditioning.

The computational cost is proportional to the number of nodes. In this algorithm, as shown in Fig. 2(a), not all the points in the cloud are considered as nodes. Instead, a voxel-grid-based down-sampling of $\mathcal{S}$ with the spacing ϵ is employed to obtain G grid points for approximation. The position of the grid points is denoted as $\mathbf{P} = \{\mathbf{p}_i^{\mathrm{g}} \in \mathbb{R}^3\}_{i=1}^{G}$. To simplify the deformation model, the radius r_i is set to be 3ϵ, with which the $\mathbf{M}$ can be guaranteed to nonsingular if the surface is complete. Regarding the incomplete surface caused by poor reconstruction or occlusion, the radius r_i should be increased until a sufficient number of grid points are included.

Therefore, according to Eq. (4), the relationship between the displacements of grid points and the manipulated points is

$$\delta\mathbf{p}_i^{\mathrm{m}} = \sum_{j=1}^{N} \phi_j(\mathbf{p}_i^{\mathrm{m}}) \delta\mathbf{p}_j^{\mathrm{g}} \tag{7}$$

where $\delta\mathbf{p}_i^{\mathrm{m}}, i \in \{1, 2, \ldots, K\}$ is the movement of the i-th manipulated point. To simplify the notation, the matrices $\delta\mathbf{p}^{\mathrm{m}} = [\delta\mathbf{p}_1^{\mathrm{m}} \quad \delta\mathbf{p}_2^{\mathrm{m}} \quad \ldots \quad \delta\mathbf{p}_K^{\mathrm{m}}]^{\intercal} \in \mathbb{R}^{K\times 3}$ and $\delta\mathbf{p}^{\mathrm{g}} = [\delta\mathbf{p}_1^{\mathrm{g}} \quad \delta\mathbf{p}_2^{\mathrm{g}} \quad \ldots \quad \delta\mathbf{p}_G^{\mathrm{g}}]^{\intercal} \in \mathbb{R}^{G\times 3}$ are introduced. The relationship between the displacement field of grid points and manipulation points is

$$\delta\mathbf{p}^{\mathrm{m}} = \mathbf{\Phi}\delta\mathbf{p}^{\mathrm{g}} \tag{8}$$

where $\mathbf{\Phi}$ is an $K \times G$ matrix whose element in i-th row j-th column is $\phi_j(\mathbf{p}_i^{\mathrm{m}})$. Our method uses this principle to generate the trajectories of the manipulators and is named the grid-point-based weighted residual method (GP-WRM).

Remark 2: Eq. (8) maps a higher dimensional field to a lower dimensional one, indicating the matrix $\mathbf{\Phi}$ has a non-trivial null space. Therefore, in this model, some grid points may not be directly affected by the movement of the manipulation points. The displacement field of the grid points, after the movement of the manipulation points is applied, is assumed to be

$$\delta\mathbf{p}^{\mathrm{g}}(\delta\mathbf{p}^{\mathrm{m}}) = \mathbf{\Phi}^{\dagger}\delta\mathbf{p}^{\mathrm{m}} : \mathbb{R}^{M\times 3} \to \mathbb{R}^{G\times 3} \tag{9}$$

where $\mathbf{\Phi}^{\dagger}$ is the pseudo-inverse of $\mathbf{\Phi}$.

Proposition 1. *Given a desired displacement field of grid point $\delta\mathbf{p}_{\mathrm{des}}^{\mathrm{g}} \in \mathbb{R}^{G\times 3}$, the Euclidean distance between desired and actual displacement field can be minimised by applying the movement of the manipulation points solved from Eq.* (8)*.*

Proof. To minimise the Euclidean distance between the desired and actual displacement field, the objective function is

$$\delta\mathbf{p}^{\mathrm{m}*} = \arg\min_{\delta\mathbf{p}^{\mathrm{m}}} \|\delta\mathbf{p}^{\mathrm{g}}(\delta\mathbf{p}^{\mathrm{m}}) - \delta\mathbf{p}_{\mathrm{des}}^{\mathrm{g}}\|_2. \tag{10}$$

The optimal solution is $\delta\mathbf{p}^{\mathrm{m}*} = \mathbf{\Phi}\delta\mathbf{p}_{\mathrm{des}}^{\mathrm{g}}$. ■

The normal of the surface at point $\mathbf{p}_i^{\mathrm{g}}$, denoted as $\mathbf{n}_i$, can be estimated from the neighbouring points on $\mathcal{S}$ by principal component analysis (PCA) [32] where the searching radius is set as ϵ.

To compute the desired displacement field from $\mathcal{S}$ to $\mathcal{T}$, as shown in Fig. 2(b), we first apply an initial rigid alignment $\mathbf{H} = \{\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3\}$ on $\mathcal{S}$, then we assume that the grid points move along the normal of surface $\mathbf{n}_i \in \mathbb{R}^3$. The initial alignment could be based on the OBB of a non-planar point cloud since the deformation is not very large. Thus, the *local* updated position of i-th grid point is $\mathbf{p}_i^{\mathrm{g}\prime} = \mathbf{R}(\mathbf{p}_i^{\mathrm{g}} + \alpha_i\mathbf{n}_i) + \mathbf{t}$ (α_i is the distance of the movement along the normal $\mathbf{n}_i$) and its normal is $\mathbf{n}_i' = \mathbf{R}\mathbf{n}_i$. The soft object would be applied a rigid transformation and then deformed locally. For every grid point $\mathbf{p}_i^{\mathrm{g}}$, a constraint exists that

$$\mathbf{u}(\mathbf{p}_i^{\mathrm{g}}) = \sum_{j=1}^{G, j\neq i} \phi_j(\mathbf{p}_i^{\mathrm{g}})\, \mathbf{u}_j. \tag{11}$$

Therefore, the constraints of all the grid points' movements can be expressed as

$$\mathbf{N}\,\mathrm{diag}(\boldsymbol{\alpha})\,\mathbf{\Phi}_{\mathrm{c}} = \mathbf{0} \tag{12}$$

where $\mathbf{N} = [\mathbf{n}_1 \quad \mathbf{n}_2 \quad \cdots \quad \mathbf{n}_i] \in \mathbb{R}^{3\times G}$, $\boldsymbol{\alpha} \in \mathbb{R}^G$ is the list of the scalar α_i, and

$$\mathbf{\Phi}_{\mathrm{c}} = \begin{bmatrix} -1 & \phi_{12} & \phi_{13} & \ldots & \phi_{1G} \\ \phi_{21} & -1 & \phi_{21} & \ldots & \phi_{2G} \\ \phi_{31} & \phi_{31} & -1 & \ldots & \phi_{3G} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{G1} & \phi_{G2} & \phi_{G3} & \ldots & -1 \end{bmatrix} \in \mathbf{R}^{G\times G}. \tag{13}$$

The element in $\mathbf{\Phi}_{\mathrm{c}}$ is $\phi_{ij} := \phi_j(\mathbf{p}_i^{\mathrm{m}})$. $\phi_j(\mathbf{p}_i^{\mathrm{m}})$ can be computed from Eq. (5), but the moment matrix is $\mathbf{M}(\mathbf{x}) = \sum_k^{G, k\neq j} \omega_k \mathbf{l}(\mathbf{x}_k)\, \mathbf{l}(\mathbf{x}_k)^{\intercal}$.

In this deformation model, the deformation gradient at position $\mathbf{x}$ on $\mathcal{S}$ is $\mathbf{J}(\mathbf{x}) = \nabla\mathbf{u}(\mathbf{x}) + \mathbf{I}_3 : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$. After the deformation, the normal $\mathbf{n}(\mathbf{x})$ is estimated to be [33]

$$\hat{\mathbf{n}}(\mathbf{x}) = \det(\mathbf{J}(\mathbf{x})) \left(\left(\mathbf{J}(\mathbf{x})\right)^{-1}\right)^{\intercal} \mathbf{n}(\mathbf{x}). \tag{14}$$

The updated normal $\hat{\mathbf{n}} \in \mathbb{R}^3$ from Eq. (14) needs to be normalised. In this study, the updated unit normal vectors of the grid point are unnecessary to be calculated. But for every manipulator point $\mathbf{p}_i^{\mathrm{m}}$, its updated surface normal, denoted as $\hat{\mathbf{n}}_i^{\mathrm{m}} := \hat{\mathbf{n}}\left(\mathbf{p}_i^{\mathrm{m}}\right)$, will be calculated for rotation, as elaborated in the Section III-C1.

Remark 3: $\mathbf{J}$ is invertible because the determinant of the deformation gradient tensor is always positive (since a negative mass density is not physically possible).

B. Target Surface

The target shape can be divided into a plane and a surface (or two-dimensional manifold).

1) Plane: In many cases, the soft object is desired to be flattened, such as for contact scanning or cloth folding. The idea is to minimise the sum of the distances of the grid points in the tangent plane of their neighbours:

$$E_{\mathrm{P}} = \frac{1}{G}\sum_{i=1}^{G}\sum_{j=1}^{G}\omega_j\left(\mathbf{p}_i^{\mathrm{g}\prime}\right)\left(\left(\mathbf{p}_i^{\mathrm{g}\prime}-\mathbf{p}_j^{\mathrm{g}\prime}\right)^{\intercal}\mathbf{n}_i'\right)^2. \tag{15}$$

In Eq. (15), we make an approximation of the updated normal as the original normal, and the scalar factor $\omega_j\left(\mathbf{p}_i'\right) = \omega_j\left(\mathbf{p}_i\right) := \omega_{ij}$, leading to a quadratic objective function:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{G}\boldsymbol{\alpha}-\mathbf{K}\|_2^2 + \lambda\|\mathbf{N}\,\mathrm{diag}\left(\boldsymbol{\alpha}\right)\boldsymbol{\Phi}_{\mathrm{c}}\|_2^2 \tag{16}$$

where $\boldsymbol{\alpha}^* = [\alpha_1^* \quad \alpha_2^* \quad \dots \quad \alpha_G^*]^{\intercal} \in \mathbb{R}^G$ is the optimum, and the scalar element in i-th row and j-th column of $\mathbf{G} \in \mathbb{R}^{G\times G}$ is

$$g_{ij} = \begin{cases} \sum_{k=1}^{G,k\neq j}\omega_{ik} & i = j \\ -\omega_{ij}\mathbf{n}_i^{\intercal}\mathbf{n}_j & i \neq j \end{cases}, \tag{17}$$

and the i-th scalar element in vector $\mathbf{K} \in \mathbb{R}^N$ is $-\sum_{j=1}^{G}\omega_{ij}\left(\mathbf{p}_i^{\mathrm{g}}-\mathbf{p}_j^{\mathrm{g}}\right)^{\intercal}\mathbf{n}_i'$.

The function in Eq. (16) can be minimised by setting it's derivative with respect to $\boldsymbol{\alpha}$ to zero:

$$\mathbf{G}^{\intercal}\left(\mathbf{G}\boldsymbol{\alpha}-\mathbf{K}\right) + \lambda\mathbf{N}^{\intercal}\mathbf{N}\,\mathrm{diag}\left(\mathbf{z}\right)\boldsymbol{\alpha} = \mathbf{0} \tag{18}$$

where the i-th element in $\mathbf{z}$ is the sum of the i-th column in $\boldsymbol{\Phi}_{\mathrm{c}}$, and the λ is the scalar parameter. The optimal solution is

$$\boldsymbol{\alpha}^* = \left(\mathbf{G}^{\intercal}\mathbf{G} + \lambda\mathbf{N}^{\intercal}\mathbf{N}\,\mathrm{diag}\left(\mathbf{z}\right)\right)^{-1}\mathbf{G}^{\intercal}\mathbf{K} \tag{19}$$

To prevent regional smoothing, The weighting function ω should have a larger radius ($5\epsilon \sim 6\epsilon$) in Eq. (15) compared to the deformation model.

If the target plane is given, defined as a normal vector $\mathbf{n}_{\mathrm{t}} \in \mathbb{R}^3$ with centre point $\mathbf{c}_t \in \mathbb{R}^3$, the matrix $\mathbf{R}$ can be updated by computing the rotation between the current normal estimated from the updated grid points via PCA and $\mathbf{n}_{\mathrm{t}}$ using minimal Riemannian distance (details in Appendix B). The translational part, $\mathbf{t}$, is the direction from the centre of the updated grid point to $\mathbf{c}_t$.

If the desired plane is not specified, meaning it is flattened locally, some parts of the soft object may not be free to move, resulting in failure of the specified plane. Some specified target planes may not be physically reachable because of external fixtures. But with that closed-loop controller, the shape can be flattened and estimated at the fixed position. In the updated model, the matrix $\mathbf{R}$ is set to be identity and vector $\mathbf{t}$ is $[0 \quad 0 \quad 0]^{\intercal}$.

2) Surface: When the target is a surface, the current surface $\mathcal{P}$ is registered to the desired surface. In a practical case, the target surface can be a mesh, a non-uniform rational B-spline (NURBS) surface [34] or a point cloud. But in this method, considering the current surface $\mathcal{S}$ is simplified with the grid points and their normal vectors, the target surface $\mathcal{T}$ is represented by the implicit MLS surface [35] as well. This method uses MLS surface representation due to its ability to approximate weighted averaging of all point-wise signed distance functions, which closely resembles the signed distance to the surface under uniform sampling conditions [35]:

$$\mathcal{T}_{\mathrm{s}}\left(\mathbf{x}\right) = \frac{\sum_{i=1}^{M}\theta_i\left(\mathbf{x}\right)\mathbf{v}_i^{\intercal}\left(\mathbf{x}-\mathbf{m}_i\right)}{\sum_{i=1}^{M}\theta_i\left(\mathbf{x}\right)} \tag{20}$$

where $\mathcal{T}_{\mathrm{s}}\left(\mathbf{x}\right) : \mathbb{R}^3 \to \mathbb{R}$ is the signed distance to $\mathcal{T}$, $\mathbf{v}_i \in \mathbb{R}^3$ is the normal vector of surface on node $\mathbf{m}_i \in \mathbb{R}^3$, the Gaussian function $\theta_i\left(\mathbf{x}\right) = \exp\left(-\|\mathbf{x}-\mathbf{m}_i\|/r_i^2\right) : \mathbb{R}^3 \to \mathbb{R}$ is set as the weight function with control radius r_i. M is the number of nodes. The implicit function of surface $\mathcal{T}$ is defined by the zero level set of Eq. (20), that is $\mathcal{T}_{\mathrm{s}}\left(\mathbf{x}\right) = 0$. Regarding the selection of the points and normal vectors, the higher number of nodes can make the implicit MLS function in Eq. (20) closer to the original surface. The nodes are uniformly sampled from meshes, NURBS surfaces or point clouds. In this scenario, $\mathcal{T}$ should be sampled with a higher resolution $\frac{1}{k}\epsilon(k>1)$ before the manipulation, and the Gaussian function's control radius is $r = k\epsilon$.

The objective function for computing the displacement of grid points on the surface $\mathcal{S}$ to minimise the squared distance to the target $\mathcal{T}$ in each iteration is defined as follows:

$$E_{\mathrm{S}}^2 = \frac{1}{G}\sum_{i=1}^{G}\mathcal{T}_{\mathrm{s}}\left(\mathbf{p}_i^{\mathrm{g}\prime}\right)^2. \tag{21}$$

To find the optimum of this non-linear objective function, the estimated vector of displacement $\hat{\boldsymbol{\alpha}}$ is updated according to the rule

$$\delta\hat{\boldsymbol{\alpha}} = -\gamma\frac{\partial E_{\mathrm{S}}^2}{\partial\boldsymbol{\alpha}} - \gamma\lambda\mathbf{N}^{\intercal}\mathbf{N}\,\mathrm{diag}\left(\mathbf{z}\right)\boldsymbol{\alpha} \tag{22}$$

for $\gamma > 0$ as a tuning gain. The $\hat{\boldsymbol{\alpha}}$ is initialised with a zero vector. The i-th element in the gradient of the Eq. (21) is

$$\frac{\partial E_{\mathrm{S}}^2}{\partial\boldsymbol{\alpha}_i} = \frac{2}{G}\mathbf{R}\mathbf{n}_i^{\intercal}\mathcal{T}_{\mathrm{s}}\left(\mathbf{p}_i^{\mathrm{g}\prime}\right)\nabla\mathcal{T}_{\mathrm{s}}\left(\mathbf{p}_i^{\mathrm{g}\prime}\right) \tag{23}$$

Due to the continuity of Eq. (20), the gradient $\nabla\mathcal{T}_{\mathrm{s}}\left(\mathbf{x}\right)$ can be analytically solved, see its expression in the Appendix C. It can also be approximated as

$$\nabla\mathcal{T}_{\mathrm{s}}\left(\mathbf{x}\right) = \frac{\sum_{i=1}^{M}\theta_i\left(\mathbf{x}\right)\mathbf{v}_i}{\sum_{i=1}^{M}\theta_i\left(\mathbf{x}\right)}, \tag{24}$$

to reduce the computational cost if there are many control points in $\mathcal{T}$. To enhance the robustness to outliers, we truncate the $\mathcal{T}_{\mathrm{s}}\left(\mathbf{x}\right)$ to a constant value at a distance of 3ϵ if some grid points are farther than 3ϵ.

The gradient descent estimator aims to determine $\boldsymbol{\alpha}^*$ and then calculate the displacement field. Finding the true optimum of the objective function is unnecessary, as the controller will repeat the process several times until the target shape is achieved. The implementation of the update rule is outlined in Algorithm 1.

Algorithm 1 Find optimal displacement to target surface

Require: $\mathcal{T}$, $\mathcal{S}$, $\mathbf{P}$, $\mathbf{n}$
Ensure: $\boldsymbol{\alpha}^*$
Initialization:
$\mathbf{R}, \mathbf{t} \leftarrow \text{InitialAlignment}\,(\mathcal{S}, \mathcal{T})$
$\boldsymbol{\alpha}^* \leftarrow \mathbf{0}_N$, $\|\delta\hat{\boldsymbol{\alpha}}\|_2 \leftarrow +\infty$
LOOP Process:
while $\|\delta\hat{\boldsymbol{\alpha}}\|$ not small enough **do**
  $\delta\hat{\boldsymbol{\alpha}} \leftarrow$ Eq. (22)
  $\boldsymbol{\alpha}^* \leftarrow \boldsymbol{\alpha}^* + \delta\hat{\boldsymbol{\alpha}}$
  Update $\mathbf{p}_i^{\mathrm{g}\prime}$
end while
return $\boldsymbol{\alpha}^*$

C. Surface-Servo Controller

1) 6-DoF Optimal Trajectory: We design a closed-loop controller to actively deform the surface of the soft object, as illustrated in Fig. 3. In each iteration, the grid points $\mathbf{P}$ with normal $\mathbf{n}$ are down-sampled from the $\mathcal{S}$. Then a new deformation model for $\mathcal{S}$ to $\mathcal{T}$ is computed according to the target shape. The desired end pose of the robots can be determined before being sent to the low-level controller. The soft object deforms as the robots move. The iteration continues until the difference between $\mathcal{S}$ and $\mathcal{T}$ (that is, the energy E_{P} or E_{S}) is smaller than a threshold.

Let the estimated end-effector Cartesian rotation of i-th robot in each iteration be $\hat{\mathbf{R}}_i$. The rotation should be updated as the normal of the grid point changes, according to Eq. (14). Therefore, there is a relation:

$$\hat{\mathbf{R}}_i \mathbf{R} \mathbf{R}_i^{\intercal} \mathbf{n}_i^{\mathrm{m}} = \hat{\mathbf{n}}_i^{\mathrm{m}} \tag{25}$$

where $\mathbf{n}_i^{\mathrm{m}} \coloneqq \mathbf{n}\left(\mathbf{p}_i^{\mathrm{m}}\right) \in \mathbb{R}^3$ is the normal vector of the i-th manipulation point. The manipulation points may not belong to the group $\mathbf{P}$, but $\mathbf{n}_i^{\mathrm{m}}$ can still be estimated by the surrounding point cloud. To minimise the rotation in Riemannian distance, the objective function is as follows:

$$d = \frac{1}{\sqrt{2}} \left\| \log\left(\hat{\mathbf{R}}_i \mathbf{R} \mathbf{R}_i^{\intercal}\right) \right\|_2 \tag{26}$$

where $\log(\cdot) : SO(3) \rightarrow \mathfrak{so}\,(3)$ is the logarithmic map from Lie group (3D rotation group) to Lie algebra. The minimised $d_{\min} = |\arccos\left(\hat{\mathbf{n}}_i^{\mathrm{m}\intercal} \mathbf{n}_i^{\mathrm{m}}\right)|$, see proof in Appendix B. Therefore, the rotation in each iteration is

$$\hat{\mathbf{R}}_i = \exp\left(d_{\min} \mathbf{v}\right) \mathbf{R}_i \mathbf{R}^{\intercal} \tag{27}$$

where $\exp(\cdot) : \mathfrak{so}\,(3) \rightarrow SO(3)$ is the exponential map, and $\mathbf{v} = \mathbf{n}_i^{\mathrm{m}} \times \hat{\mathbf{n}}_i^{\mathrm{m}}$ is a normalised vector.

Therefore, the 6-DoF movement of K robots in each iteration is $\delta\mathcal{R} = \left\{ \left(\hat{\mathbf{R}}_i^{\mathrm{m}} \mathbf{R}_i^{\mathrm{m}\intercal}, \delta\mathbf{p}_i^{\mathrm{m}}\right) \right\}_{i=1}^{K}$.

2) Velocity Controller: The robots are controlled by velocity in joint space. Since the robots' movements in initial iterations are largely due to the significant shape difference between current and target surfaces, a velocity cap is set for safety reasons. Thus the velocity $\dot{\mathcal{R}} \in \mathbb{R}^6$ in Cartesian space is as

$$\dot{\mathcal{R}} = \mathbf{U}\left(\frac{1}{\Delta T}\begin{bmatrix} \delta\mathbf{p}_i^{\mathrm{m}} \\ \log\left(\hat{\mathbf{R}}_i^{\mathrm{m}} \mathbf{R}_i^{\mathrm{m}\intercal}\right) \end{bmatrix}\right) \tag{28}$$

where ΔT is the time interval of iteration. The cap function is $\mathbf{U}(\mathbf{x}) = [U_1(\mathbf{x}_1) \quad U_2(\mathbf{x}_2) \quad \dots \quad U_6(\mathbf{x}_6)]^{\intercal} : \mathbb{R}^6 \rightarrow \mathbb{R}^6$, where the scalar function

$$U_i(x) = \begin{cases} x, & |x| < m_i \\ m_i \operatorname{sgn}(x), & |x| \geq m_i. \end{cases} \tag{29}$$

The positive scalars m_i determine the maximum velocities during manipulations. The velocity in joint space $\dot{\mathbf{q}}$ is as

$$\dot{\mathbf{q}} = \left(\frac{\partial\mathcal{R}}{\partial\mathbf{q}}\right)^{-1} \dot{\mathcal{R}} \tag{30}$$

where $\partial\mathcal{R}/\partial\mathbf{q}$ is the Jacobian matrix.

D. Convergence and Stability Analysis

Assuming the target is physically reachable, the energy function (E_{P} or E_{S}) is asymptotically minimised.

Proposition 2. *The movements of the manipulators will asymptotically converge to zero if the target shape is physically reachable.*

Proof. Both the E_{P} and E_{S} are μ-strong convex and differentiable. Here, we define a unified energy function $E(\mathbf{P}) : \mathbb{R}^{3N} \rightarrow \mathbb{R}$. Let the $\delta\mathbf{P}^* \in \arg\min_{\boldsymbol{\alpha}} E(\mathbf{P})$, for any $\mathbf{P}$:

$$E(\mathbf{P}) - E(\mathbf{P}^*) \geq \frac{\mu}{2} \|\mathbf{P} - \mathbf{P}^*\|_2^2 \tag{31}$$

where $\mu > 0$. When $E(\mathbf{P}) - E(\mathbf{P}^*) \rightarrow 0$, thus $\mathbf{P} \rightarrow \mathbf{P}^*$, which means $\sum_i^N \|\delta\mathbf{p}_i^{\mathrm{g}}\|_2 \rightarrow 0$.

According to Eq.(8), the 2-norm of the displacements on the manipulation points is

$$\|\delta\mathbf{p}_i^{\mathrm{m}}\|_2 = \left\| \sum_i^N \phi_i\left(\mathbf{p}_i^{\mathrm{m}}\right) \delta\mathbf{p}_i^{\mathrm{g}} \right\|_2 \leq \sum_i^N \left|\phi_i\left(\mathbf{p}_i^{\mathrm{m}}\right)\right| \|\delta\mathbf{p}_i^{\mathrm{g}}\|_2 \tag{32}$$

Because $\sum_i^N \|\delta\mathbf{p}_i^{\mathrm{g}}\|_2 \rightarrow 0$, $\delta\mathbf{p}_i^{\mathrm{m}} \rightarrow 0$. This proves the asymptotic convergence of the movements of robots. ■

To analyse the stability of the control law, Eq. (28) shows that it is not a dynamic system; therefore, this high-level control system is stable.

IV. Simulation Validation

Simulation experiments were conducted using the SOFA simulator [36], as shown in Fig. 4 and the accompanying video 1. The performance of the adaptive Jacobian-based method for soft object manipulation [11] and the data-driven method (Fast Online Gaussian Process Regression, FO-GPR) [22] were evaluated on a 3D liver model. The physical model was a triangular FEM model with a Poisson ratio of 0.3 and Young's

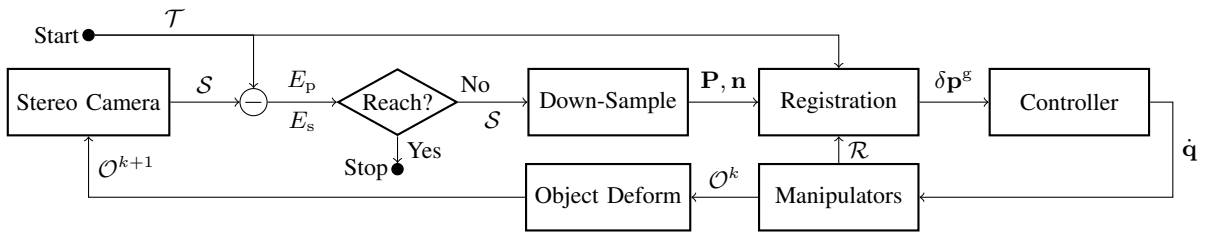


Fig. 3. Schematic representation of the proposed controller. $\mathcal{O}^k$ and $\mathcal{O}^{k+1}$ are the configurations of the soft object in k-th and $(k+1)$-th iterations, respectively.

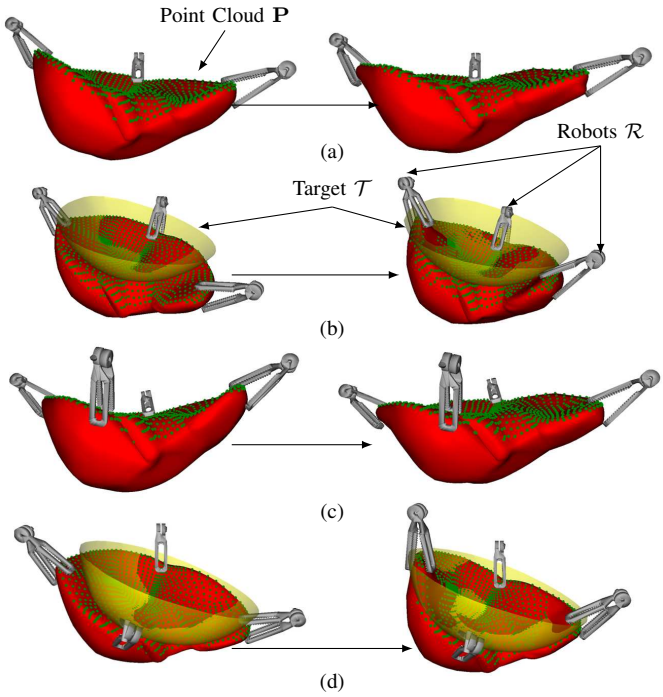


Fig. 4. Simulated robotic manipulation of a soft object. On the left is the initial setup, and on the right is the final result in each sub-figure. The green dots represent the point cloud from the stereo camera. The yellow surfaces are the targets in (b) and (d).

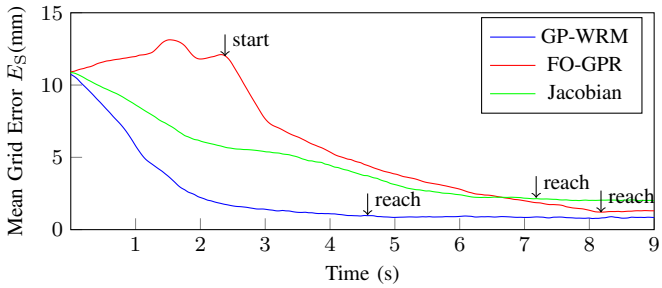


Fig. 5. Mean grid errors E_{S} of the manipulation experiments (Task 4) with different methods: FO-GPR [18] and Jacobian-based [11].

modulus of 500 $\mathrm{N\,m^{-2}}$. Three or four manipulation points with applied friction were used to simulate grasping force. The point cloud consisted of the vertices on the mesh model facing the camera. In the simulation, five vertices on the mesh model were selected as feature points and used in Jacobian-based and FO-GPR methods. The position and shape terms were calculated using these feature points, and the surface variation was computed using the point cloud around the feature points. In the SOFA simulation environment, all vertices' positions and normal vectors can be obtained directly.

Fig. 5 demonstrates a decrease in shape errors for all three controllers. The shape error is defined as the mean grid error in all three methods. While grid points are unnecessary for Jacobian-based and FO-GPR, grid points at the exact down-sampling resolution as ours are still used in these two methods to measure the mean grid errors. Table I summarises simulation results with different configurations. In Task 1 and 2, the target surface is defined as a plane, while in Task 3 and 4, it is a curved surface. Each task was implemented 5 times with various targets. The number of manipulation points (K) was varied between 3 in Task 1 and 3, and 4 in Task 2 and 4 to examine its effect. The mean grid error and time were recorded for each method. The controller with more manipulation points was observed to perform better, achieving the target shape more easily. These findings are consistent with the results shown in Fig. 5.

The Jacobian-based method is less accurate by 45.6% compared to ours because it uses only a limited number of points as the states to track the deformation during manipulations. The shape error of FO-GPR initially increases during the random exploration to learning the model parameters and finally reaches a minimal value of 15.9% higher than our method. Regarding the manipulation time, the Jacobian-based controller uses the adaptive deformation model to linearise the system, while the non-linear methods (GP-WRM and FO-GPR) establish a non-linear relationship between the state and action space, resulting in faster convergence of shape error. Compared to the linear method, our method saves 55.2% of manipulation time. However, the data-driven method (FO-GPR) requires additional online exploration to gather sufficient data to learn the parameters in the non-linear model, which can take an extra approximately 2s in our experiments. The manipulation time for the data-driven approach is around 25.7% longer than in our method.

V. Experimental Validation

In this section, we validate the proposed method through real-world experiments.

A. Setup

The experimental setup is shown in Fig. 6. It involves using the *da Vinci* Surgical System (Intuitive Surgical, USA)

TABLE I
THE PERFORMANCE OF THREE METHODS IN SHAPE CONTROL

Task	Target	K	Methods	Mean Error(mm)	Time(s)
1	Plane	3	GP-WRM	1.23 ± 0.32	3.39 ± 0.25
			Jacobian	1.83 ± 0.52	5.21 ± 0.68
			FO-GPR	1.53 ± 0.67	5.81 ± 0.77
2	Plane	4	GP-WRM	0.93 ± 0.22	3.18 ± 0.53
			Jacobian	1.61 ± 0.43	5.02 ± 0.41
			FO-GPR	1.43 ± 0.30	5.58 ± 0.49
3	Surface	3	GP-WRM	1.63 ± 0.61	4.98 ± 0.49
			Jacobian	2.33 ± 0.73	7.57 ± 0.61
			FO-GPR	1.87 ± 0.29	8.81 ± 0.39
4	Surface	4	GP-WRM	1.39 ± 0.46	4.67 ± 1.03
			Jacobian	1.98 ± 0.87	7.25 ± 1.07
			FO-GPR	1.53 ± 0.57	8.16 ± 0.94

Fig. 6. The setup of the experiment using the *da Vinci* Surgical System (only two PSMs were used), a stereo camera and a soft object (Phantom colon).

equipped with two 7-DOF patient-side manipulators (PSM) for validation. The surgical instruments with wristed dexterity can grip soft objects. The hardware controller is the *da Vinci* Research Kit (dVRK) [37]. An RGBD 3D camera (RealSense D435i, Intel, USA) was utilised to obtain a high-precision point cloud. Robotics control was facilitated by the Robot Operating System (ROS), while the Point Cloud Library (PCL) [38] was used to filter the point cloud, compute the surface normal vector field, and down-sample.

B. Shape Control with Various Materials

Three soft materials (fabric, paper, and silicone elastomer sheet) with a size of $10\mathrm{cm}\times5\mathrm{cm}$ were chosen to evaluate the performance of the controller with objects of different stiffness. They were grasped by the two PSMs at similar positions, as shown in Fig. 7. The experiments were divided into six groups, three with target shapes as planes and three with curved surfaces. Each group used the same target shapes for different soft objects. The target planes are defined with the $\mathbf{c}_{\mathrm{t}}$ and $\mathbf{n}_{\mathrm{t}}$ directly. The target surfaces are illustrated in the images. The tuning gain was set as $\gamma = 500$. The down-sample resolution was $\epsilon = 15\mathrm{mm}$, resulting in the grid point size of $N = 50 \pm 10$ (N varies during deformation). The maximum linear speed in Eq. (29) was set to $10\mathrm{mm/s}$, and angular speed was set to $0.5\mathrm{rad/s}$. The frequency of the iteration was $20\mathrm{Hz}$.

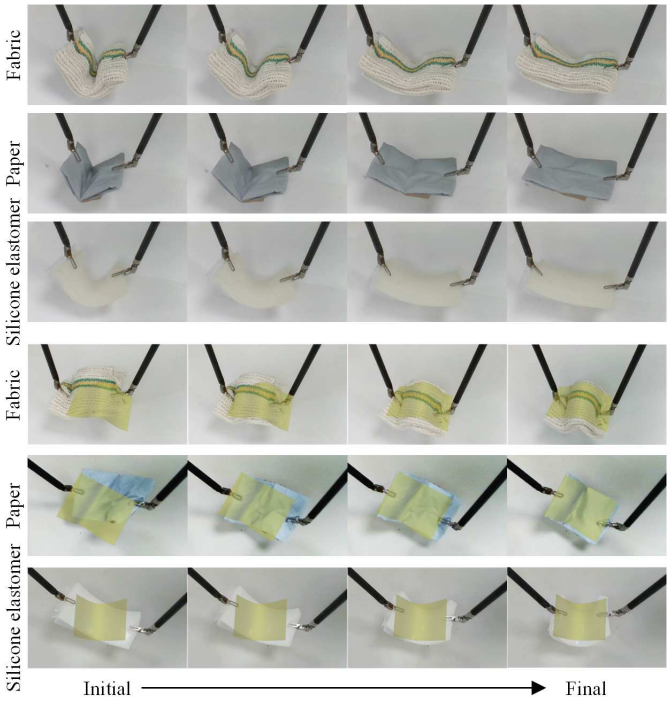


Fig. 7. Robotic manipulation experiments were conducted on fabric, paper, and silicone elastomer sheets. The first three rows show trials with planar target shapes, while the last three rows depict trials with surface target shapes. Each column showcases the frames in a time sequence from the initial to the final shape. The yellow surfaces in each trial represent the targets.

Fig. 7 displays the initial, intermediate, and final images captured from the RGB channel of the stereo camera during a single trial with a specific material. The target surfaces are superimposed on the RGB images. These figures demonstrate that the proposed method can synchronously deform soft objects and modify their poses. The objects remain unstretched during the manipulation process, while both manipulators move simultaneously. The accompanying video 2 displays all the groups of experiments.

Fig. 8 shows the mean grid error (E_{P} and E_{S}) of each trial in Fig. 7. The approach effectively reduces errors during manipulation, showing it can deform soft objects of different materials into target shapes. The error minimisation is attributed to the noise in the point cloud and the unevenness of the soft object. The initial mean error differences arise from varying initial shapes among groups. These sources of error also result in slight variations in the final Cartesian poses of the same end-effectors in the camera frame for experiments with identical shape configurations. The duration of robotic manipulations is similar when the initial and target shapes are the same across different groups.

Our proposed method demonstrates the successful robotic manipulation of soft objects to reach planar targets. However, additional challenges arise when it comes to non-planar targets, especially with soft objects made of fabric or paper. The limitations of using only two manipulators become evident as gravity, compliance, and stretchability affect the object's behaviour. The controller described in Eq. (8), which maps a higher-dimensional space to a lower-dimensional one, has

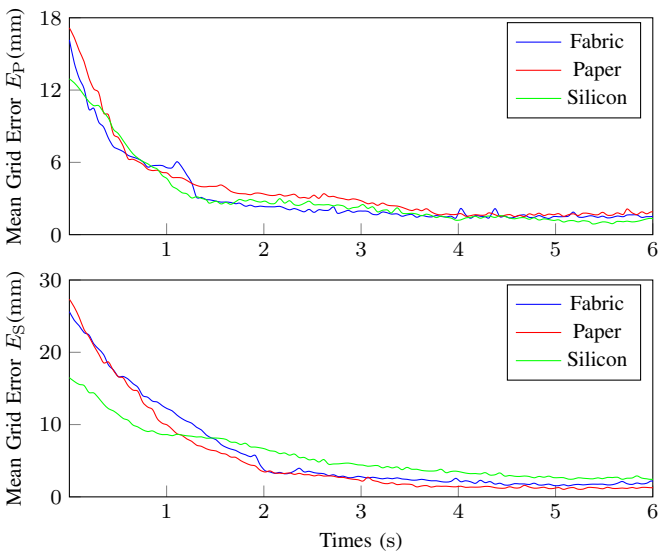


Fig. 8. Errors of the manipulation experiments for various materials. Top: the mean error E_{P} of the experiment in Fig. 7. Bottom: the mean error E_{S} of experiment in Fig. 7.

a non-trivial null space. This implies that the movements of the manipulation points may not directly impact certain grid points' motions. Additionally, the method does not consider physical forces, such as gravity, which may impact the movement of areas distant from the manipulation points. The compliance and stretchability of the material may also impede the desired movement from being realised. One way to address these challenges is by using more manipulation points, which would allow for greater control over the movement of the material, as illustrated in the simulation experiments.

C. Shape Control in RAMIS

Our method was tested on cadaveric intestine specimens to evaluate its performance in RAMIS procedures. The intestine is frequently manipulated during colorectal RAMIS procedures, such as during laparoscopic repair of small bowel obstructions or laparoscopic sigmoidectomy for colon cancer removal [7]. In these scenarios, the surgeon must handle the colon to explore or perform scans with additional diagnostic devices (e.g. ultrasound probes). Both phantom and cadaveric colons were used in the experiments. To imitate a clinical setting, part of the phantom colon was fixed to the table to simulate its attachment to the abdomen via the mesenteries.

During intracorporeal diagnostic scanning in RAMIS [39], the colon is typically grasped, and its surface is flattened to facilitate ultrasound probe scanning, as depicted in Fig. 9(a). The target plane is not predefined, and the tissue is expected to deform locally. However, folds on the tissue could result in an unclear surface representation, especially when using low-resolution grid points. To quantify the disparity between the target and actual shapes, the actual shape error is defined as:

$$E_{\mathrm{a}}^2 = \frac{1}{N}\sum_{i=1}^{N} \mathcal{T}^2\left(\mathbf{p}_i\right) \tag{33}$$

where $\mathcal{T}(\cdot) : \mathbb{R}^3 \to \mathbb{R}$ is the distance to the target surface. From Fig. 10, we can observe that the actual shape error varies as the resolution changes in each manipulation, even though they start with a similar initial shape configuration $\mathcal{O}$. To investigate the effect of down-sampling, we tested different values for the resolution ϵ, varying from 8 mm to 29 mm. Fig. 10 illustrates that a higher resolution in the down-sampled point cloud can achieve a better surface match, as it retains more surface shape information. The downside is an increased computational cost. Therefore, a balance must be struck between surface matching quality and the time to plan the robot's pose.

In laparoscopic explorations, the intention is to manipulate and deform the intestine into different shapes to reveal visually obstructed regions below the tissue. In this experiment, considering the tube-like nature of the colon, we only used a cylinder and bent tube as target surfaces. The intestinal segment between two grippers is required to deform, as shown in Fig. 9(a). The point cloud of the other part was filtered. With a resolution of $\epsilon = 20\mathrm{mm}$, the errors of these manipulations are presented in Fig. 9(b). The accompanying video 3 showcases the performance of the proposed approach with several different shapes.

D. Shape Control with Occlusion

In RAMIS procedures, the surface of the soft tissue may be occluded by the surgical instruments, and there may be missing parts due to imperfect 3D point cloud reconstruction. To evaluate the robustness of the proposed method in the presence of occlusions or poor reconstructions, several trials with simulated removal of partial point cloud on phantom colon were carried out, as shown in Fig. 11(b). The grid points in the intact point cloud are approximately 180 in each iteration.

We introduced virtual occlusions in the form of box-cropped point clouds. The boxes with the size of $10\mathrm{mm}\times10\mathrm{mm}$ were fixed in the camera frame during each operation, as shown in Fig. 11(a). We evaluated the occlusion's scatter and area. For each robot, 100 boxes were used with a position on the projection plane based on an independent 2D normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, as shown in Fig. 11(b). The mean was $\boldsymbol{\mu} = \begin{bmatrix}\mu_0 + x \\ \mu_0 + y\end{bmatrix}$, where x and y are the horizontal and vertical coordinate of the $\mathbf{p}_i^{\mathrm{m}}$, and the variance was $\boldsymbol{\Sigma} = \begin{bmatrix}\sigma_0^2 & 0 \\ 0 & \sigma_0^2\end{bmatrix}$. The distance between the occlusion centre and the manipulation point was $\sqrt{2}|\mu_0|$. The occluded area percentage, denoted as R, was calculated by considering the scatter and area of the occlusions. If some boxes overlap, the total area decreases. A positive correlation exists between the variance σ and R, as further explained in Appendix D.

From Fig. 11(c), we can find that when $|\mu_0|$ decreases, indicating that the occlusion is closer to the manipulation points, the controllers are more likely to fail, particularly in the case of continuous and larger occlusions. The failure occurs because when surrounding grid points are missing, the radius r must be increased to include enough points for displacement approximation. Conversely, a higher variance σ_0^2

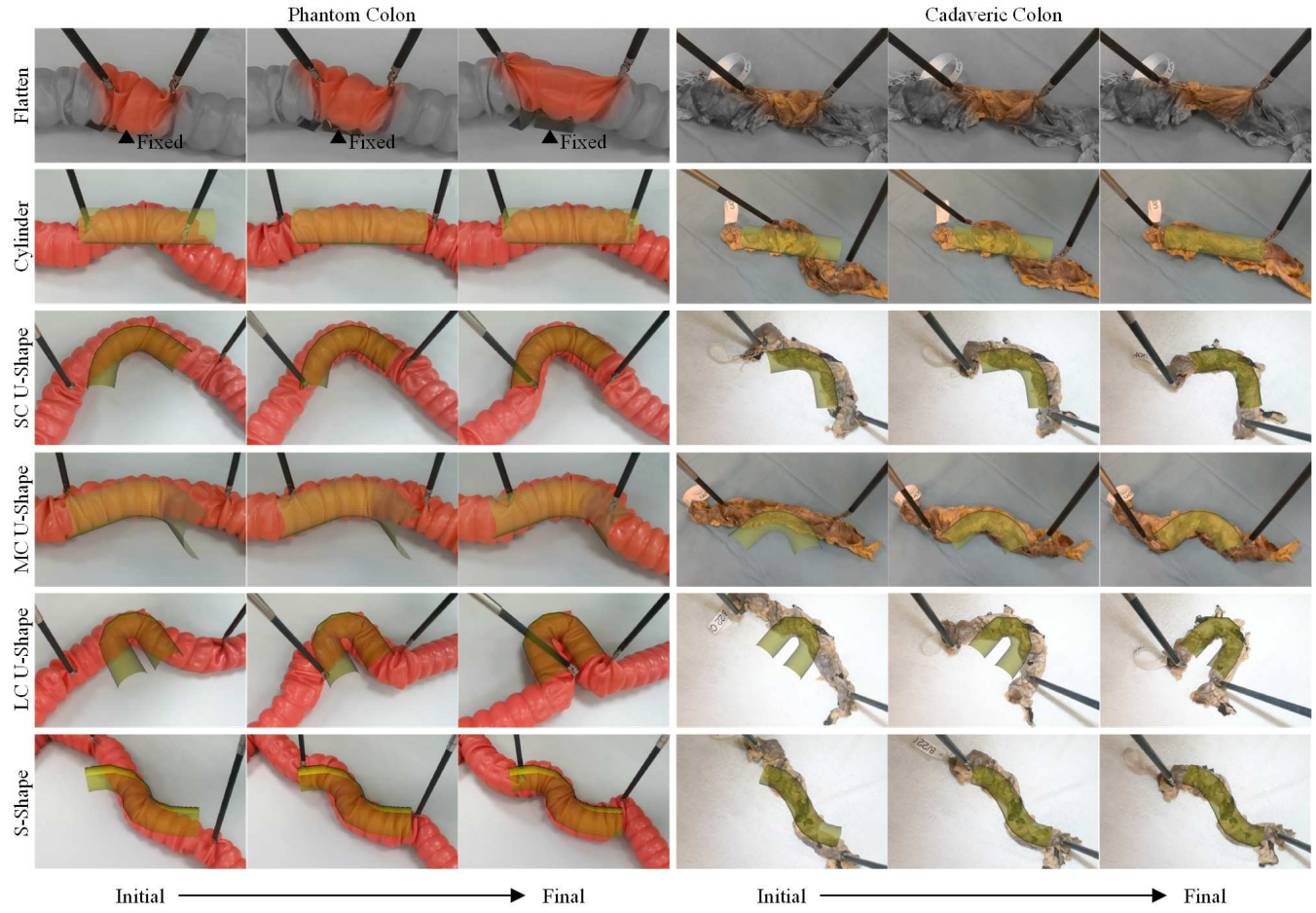


Fig. 9. (a) Results of the phantom (left) and cadaveric (right) experiments on intestine manipulation. Each column includes the frames in time sequence from initial to final shape. The target is flattening the segment between two PSMs in the first row, highlighted in the figures. The target plane is not specified. A part of the phantom is fixed to the table. In both trials, the down-sampling resolution is $\epsilon = 10$ mm. In the second row, the target surface is part of a cylinder. From the third to fifth rows, the target surface is part of a bent tube with different curvatures (LC: large curvature, MC: middle curvature, SC: small curvature). In the sixth row, the target surface is part of a S-shape tube. The yellow surfaces represent the targets in the second to sixth rows. (b) Errors of shape control on phantom (left) and cadaveric (right) colon.

leads to better performance. Even if small parts of the point cloud are missing, the grid points can still effectively represent the 3D surface of the soft object.

Compared to the occlusion removing algorithm proposed by Hu *et al*. [22], which utilises a deep neural network for vision reconstruction, our method achieves the desired shape without requiring surface reconstruction. In [25], selecting sample points or contours becomes necessary when occlusions occur. Instead, we adjust the deformation model's radius and the down-sampling filter's resolution to overcome occlusions.

The cadaveric experiment reached the target surface even with approximately 30% occlusion caused by one of the manipulators, as shown in Fig. 12(a) and the accompanying video 3. Fig. 12(b) shows the errors of these manipulations.

VI. Discussion and Conclusion

In this study, we propose a novel approach for shape control of soft objects with 3D feedback. Our method includes a

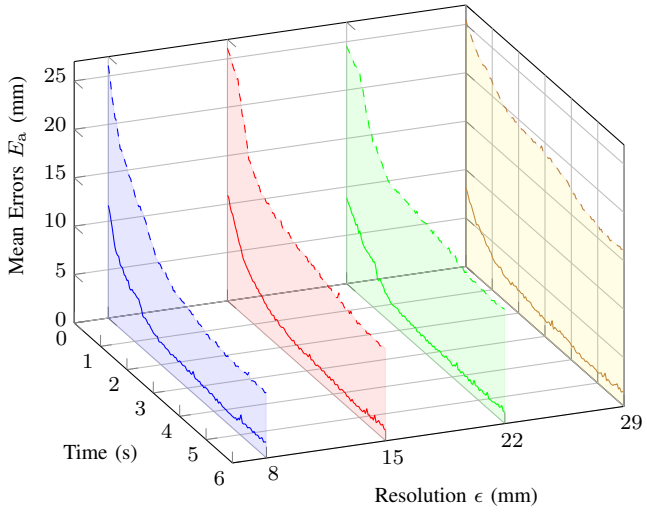
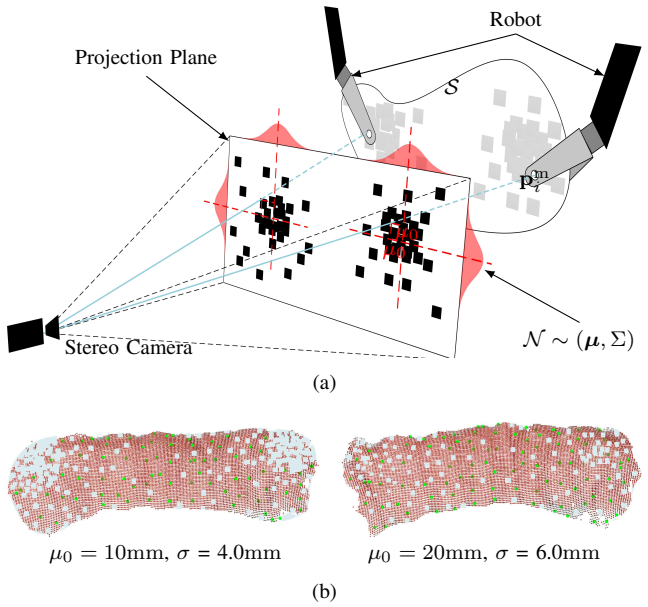


Fig. 10. Flattening phantom colon with different resolution ϵ. The curves represent the mean error E_P, and the dashed curves are for the mean actual shape error E_a.



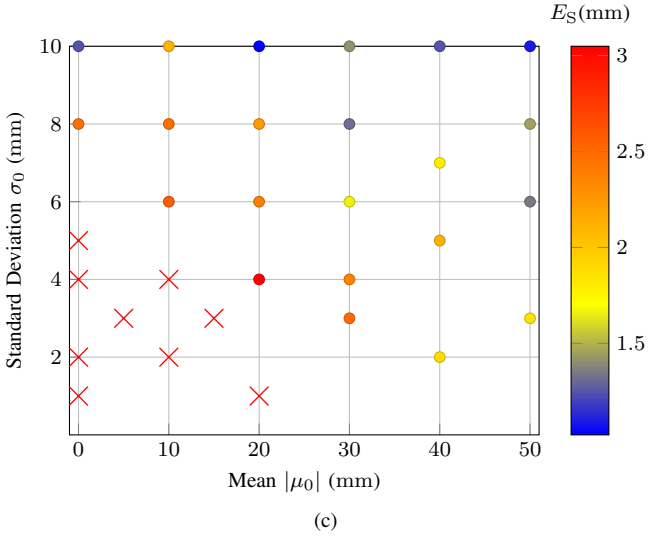


Fig. 11. (a) The illustration of different occlusion distributions. Black squares on the projection plane represent virtual boxes; red curves represent 2D normal distributions of these boxes; the grey area on surface $\mathcal{S}$ indicates occluded areas removed from the point cloud, and blue dashes are projection lines from camera to manipulation points. (b) Two examples of blocked surfaces. Green dots are grid points; grey regions indicate occlusions. (c) Results of our method with various occlusions. Data marked with cross symbols indicate targets that could not be achieved.

WRM-based deformation model for estimating gripper movements using the displacement field of surrounding points. We explain the computation of the displacement field for two types of shapes. Additionally, the approach includes a 6-DoF optimal trajectory and velocity controller. The method's effectiveness is demonstrated through simulation and experimental results.

As demonstrated in the results of our simulated experiments, this approach offers a universal solution without selecting latent representations, detecting explicit features, or creating a physical deformation model. The method reduces the computational cost by representing the real-time surface with grid points and not requiring demonstration data or mechanical properties before manipulation [10], [14], [15]. It optimises the displacement field of grid points to estimate the movement of robotic manipulators, with the computation burden mainly arising from constructing and inverting matrices $\mathbf{M}$. While the gradient descent method for surface shape control may require some time, the computational cost is still lower than that of other methods. The most significant reason for the improvement in efficiency may be attributed to the fact that, in the mapping from shape differences to robot movements, the Jacobian-based methods [12], [11], [20], [25] generate a roughly approximated linearization, necessitating iterative optimisations to estimate the Jacobian matrix. The learning-based methods [22], [17] requires online or offline exploration to update the parameters in the non-linear mapping model. The physics-based methods, such as FEM, necessitates the 3D modelling, discretisation, and the determination of the stiffness matrix of the soft object before manipulation. Additionally, they require force estimation and solving balancing functions during the manipulation process, incurring high computational costs or addition exploration to simulate a high-fidelity physical model [14]. In contrast, our method employs the WRM to establish a direct non-linear mapping based on surface geometric consistency. In terms of accuracy, the grid-point-based representation is closer to the real surface than the feature-based approach [11] or implicit [22] representation. Therefore, the difference between the current and the target surface, which is aimed to be reduced, may be more precise in our method.

The experiments involving different materials and RAMIS showcase our method's capability to achieve shape control on various soft objects with different mechanical properties. The success in reaching different target shapes during the RAMIS experiments demonstrates the suitability of our method for these surgical subtasks. Some failed cases demonstrate that external forces, compliance, and stretchability may influence the performance of our method. The limited number of manipulators also leads to situations where certain target surfaces cannot be reached. The simulated experiments reveal that

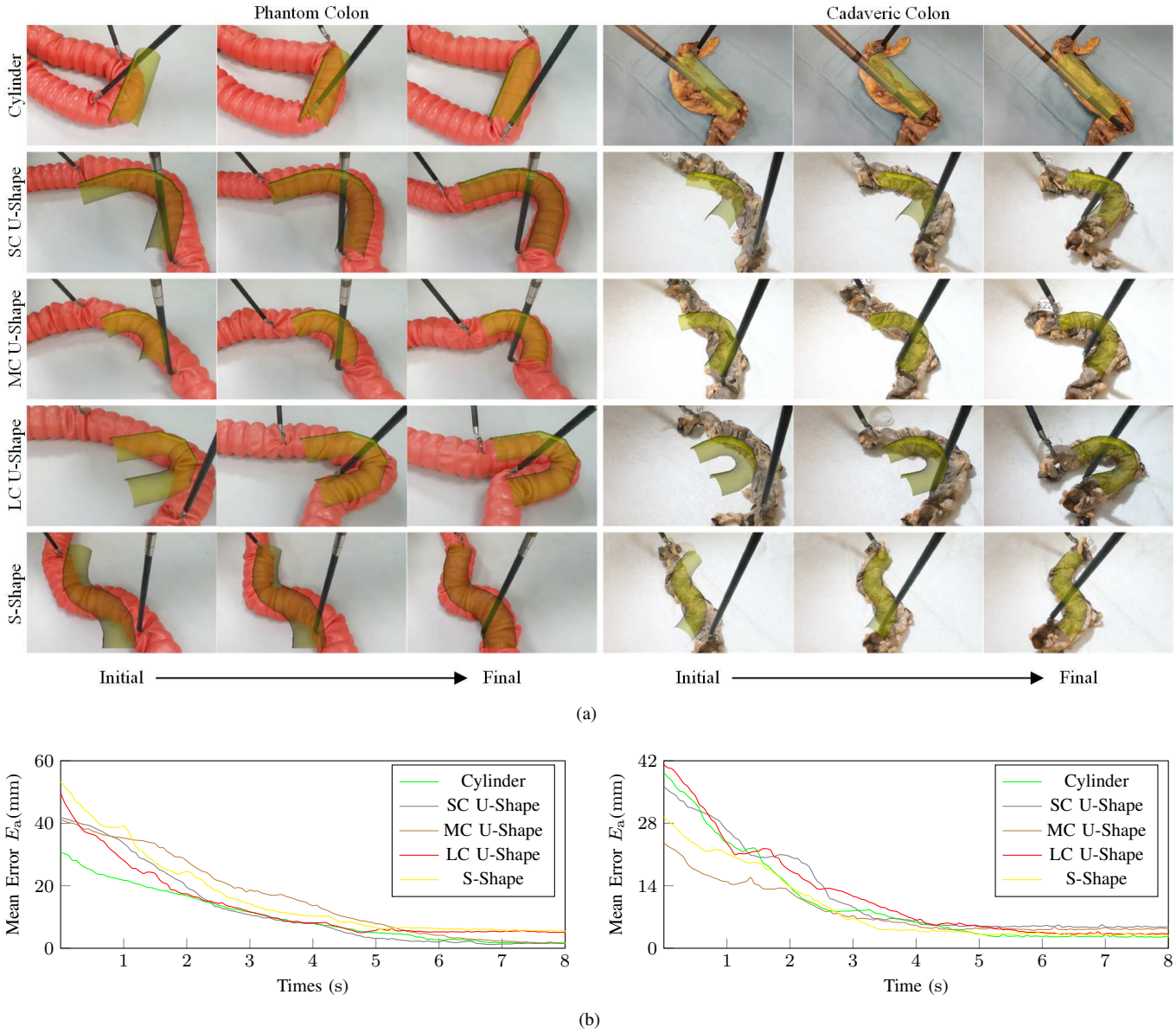


Fig. 12. (a) Results of phantom (left) and cadaveric (right) experiments on intestine manipulation with partial occlusion by the surgical instrument. Each column displays frames in a time sequence from the initial to the final shape. The yellow surfaces represent the targets. In the first row, the target surface is part of a cylinder. From the second to fourth rows, the target surface is part of a bent tube with different curvatures (LC: large curvature, MC: middle curvature, SC: small curvature). In the fifth row, the target surface is part of an S-shaped tube. (b) Errors of shape control on phantom (left) and cadaveric (left) colon with occlusion.

the 4-arm system exhibits higher controllability over the soft object than the 3-arm system. Our method is also robust against occlusions, as an accurate surface representation using grid points can be ensured by appropriately adjusting the down-sampling resolution, which can be easily achieved by adjusting the down-sample radius. Furthermore, our controller has the added benefit of producing 6-DoF Cartesian posture or velocity of the end-effector, while other works mainly focus on 3-DoF position control. This method has limitations, such as potential failure with excessive deformation or improper resolution in the down-sampling filter. It is also unsuitable for tasks where points or lines represent the target surface.

There are some limitations to our approach. First, in cases where the deformation is significant, reaching the target shape may not be possible, especially if the soft object needs to be turned over. This is because the registration of the current surface $\mathcal{S}$ to the target shape $\mathcal{T}$ may not be effective, as $\mathcal{S}$ only represents the surface facing the camera and corresponds to the same part of the object as $\mathcal{T}$. Therefore, the desired deformation should be interpolated into a series of sequential target shapes in such cases. Another issue is that if the resolution of the down-sampling filter is not appropriate, the grid points may not represent the object's actual shape exactly. This can result in inaccuracies in the method's performance, especially when surface parts are occluded.

Our method has limitations regarding explicit feature application as the limited number of feedback points may not guarantee the singularity of the moment matrix. However,

these features can still improve registration accuracy, as demonstrated in [40], where geometric features were incorporated into the error function for more precise point cloud registration.

In future directions, integrating force sensors on the grippers is envisioned to enrich the system's information. Furthermore, a simultaneous shape-planning approach for handling complex deformations by dividing them into a series of small target shapes is being explored to achieve higher levels of autonomy. Research on determining optimal manipulation points before object grasping will also be pursued.

APPENDIX A
PROOF OF THE NON-SINGULARITY OF MOMENT MATRIX

Proposition 3. *The matrix* $\mathbf{M}$ *is non-singular only when at least four non-plane points exist.*

Proof. Since the moment matrix $\mathbf{M}$ is composed of some weighted projection matrices in four-dimension, it can be expressed like

$$\mathbf{M}(\mathbf{x}) = \sum_i \lambda_i \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\intercal \tag{34}$$

where $\boldsymbol{\beta}_i$ is a normalised vector of $\mathbf{q}(\mathbf{x}_i)$ in Eq. (6), $\lambda_i = \omega_i \|\mathbf{q}(\mathbf{x}_i)\|$ is a coefficient. Given any vector $\boldsymbol{\gamma} \in \mathbb{R}^4$, $\mathbf{M}\boldsymbol{\gamma}$ is the combination of weighted projected $\boldsymbol{\gamma}$ on different vectors $\boldsymbol{\beta}_i$, hence there exists

$$\mathbf{M}\boldsymbol{\gamma} = \sum_i \lambda_i b_i \boldsymbol{\beta}_i \tag{35}$$

where $b_i = \boldsymbol{\beta}_i^\intercal \boldsymbol{\gamma}$ is the projected length. Thus, $\mathbf{M}\boldsymbol{\gamma}$ is the linear combination of the vector space $V : \{\boldsymbol{\beta}_i\}$, as well as the column space of $\mathbf{M}$. If $\mathbf{M}$ is full-rank, the dimension of the subspace of V is 4. Therefore, there are at least 4 bias vectors.

In the deformation model, at least 4 non-plane points in the surrounding ensure the non-singularity. ■

If the $\mathbf{M}$ is in an ill condition, the surface region is close to flat. The displacement of that point can be estimated with the weighted mean of surrounding displacements; that is, the scalar shape function is as follows:

$$\phi_i(\mathbf{x}) = \omega_i(\mathbf{x}) / \sum_{j=1}^{N} \omega_j(\mathbf{x}). \tag{36}$$

APPENDIX B
PROOF OF THE MINIMAL ROTATION

Proposition 4. *Given two 3D non-parallel unit vectors* $\mathbf{e}_1$ *and* $\mathbf{e}_2$*, the rotation matrix can transfer* $\mathbf{e}_1$ *to* $\mathbf{e}_2$ *with minimal Riemannian distance* $d_{\min} = \mathbf{e}_1^\intercal \mathbf{e}_2$ *is* $\exp(d_{\min}\mathbf{e}_1 \times \mathbf{e}_2)$.

Proof. Riemannian distance d represents the arc length of the shortest geodesic curve (great-circle arc), as well as the angle of the rotation matrix in axis-angle representation. According to the definition of rotation angle, $d = \arccos(\mathbf{e}_1 \times \mathbf{e})^\intercal (\mathbf{e} \times \mathbf{e}_2)$, where $\mathbf{e} \in \mathbb{R}^3$ is the unit rotation vector satisfying $(\mathbf{e}_1 - \mathbf{e}_2)^\intercal \mathbf{e} = 0$. The rotation angle can also be expressed as

$$\begin{aligned} d &= \arccos \frac{(\mathbf{e}_1 - \mathbf{e}_1^\intercal \mathbf{e}\mathbf{e})^\intercal (\mathbf{e}_2 - \mathbf{e}_2^\intercal \mathbf{e}\mathbf{e})}{\|\mathbf{e}_1 - \mathbf{e}_1^\intercal \mathbf{e}\mathbf{e}\| \|\mathbf{e}_2 - \mathbf{e}_2^\intercal \mathbf{e}\mathbf{e}\|} \\ &= \arccos \frac{\mathbf{e}_1^\intercal \mathbf{e}_2 - (\mathbf{e}_1^\intercal \mathbf{e})^2}{1 - (\mathbf{e}_1^\intercal \mathbf{e})^2}, \end{aligned} \tag{37}$$

which is monotonically increasing since $(\mathbf{e}_1^\intercal \mathbf{e})^2 \in [0, 1)$. Therefore, the minimal Riemannian distance is $d_{\min} = \mathbf{e}_1^\intercal \mathbf{e}_2$ when both $\mathbf{e}_1$ and $\mathbf{e}_2$ are vertical to the rotation vector, that is $\mathbf{e} = \mathbf{e}_1 \times \mathbf{e}_2$. The rotation matrix is $\exp(d_{\min}\mathbf{e}_1 \times \mathbf{e}_2)$, converted from the axis-angle representation by Rodrigues' formula. ■

APPENDIX C
THE EXPRESSION OF GRADIENT OF THE MLS SURFACE

$$\begin{aligned} \nabla \mathcal{T}_{\mathrm{s}}(\mathbf{x}) = &\left(\sum_{i=1}^{M} \theta_i(\mathbf{x}) \right)^{-2} (\\ &\sum_{i=1}^{M} \theta_i'(\mathbf{x}) \sum_{i=1}^{M} \theta_i(\mathbf{x}) \mathbf{v}_i^\intercal (\mathbf{x} - \mathbf{m}_i) + \\ &\sum_{i=1}^{M} \theta_i(\mathbf{x}) \sum_{i=1}^{M} (\theta_i'(\mathbf{x}) \mathbf{v}_i^\intercal (\mathbf{x} - \mathbf{m}_i) + \theta_i(\mathbf{x}) \mathbf{v}_i)) \end{aligned} \tag{38}$$

where $\theta_i'(\mathbf{x}) = -2(\mathbf{x} - \mathbf{m}_i)\left(\|\mathbf{x} - \mathbf{m}_i\| / r_i^2\right) \theta_i(\mathbf{x})$.

APPENDIX D
THE EXPECTED VALUE OF THE OCCLUSION AREA

Since M boxes are independent of each other, the expected value of the occlusion area is

$$S(\sigma) = \iint_{\mathbb{R}^2} \left(1 - (P_X + P_Y - P_X P_Y)^M\right) \mathrm{d}x\mathrm{d}y \tag{39}$$

where $P_X = P(X \geq x + b) + P(X \leq x - b)$ and $P_Y = P(Y \geq y + b) + P(Y \leq y - b)$. $P(X > x)$ is the probability of the normal distribution $\mathcal{N}(\mu, \sigma^2)$, and b is the half side length of the box. The P_X and P_Y can be computed using the normal cumulative distribution function: $P_X = 1 - \Phi\left(\frac{x+b}{\sigma}\right) + \Phi\left(\frac{x-b}{\sigma}\right)$ and $P_Y = 1 - \Phi\left(\frac{y+b}{\sigma}\right) + \Phi\left(\frac{y-b}{\sigma}\right)$.

Here, we use a numerical method to estimate the relationship between the area S and the variance σ^2, as shown in Fig. 13.

ACKNOWLEDGMENT

All the experiments involving human cadaveric tissues were performed under ethical approval from the University of Leeds. We thank the STORM lab technician, Samwise Wilson, for his hardware support, and the anatomy facilities technicians, Sarah Wilson and Charlotte Coleman, for their support in the cadaveric experiment.

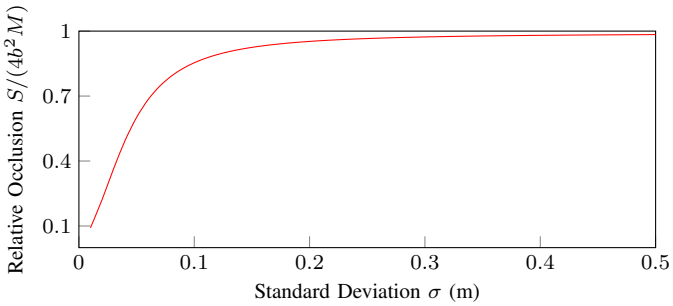


Fig. 13. The relation between the expected occlusion area and the standard deviation.

REFERENCES

[1] B. Jia, Z. Pan, Z. Hu, J. Pan, and D. Manocha, "Cloth manipulation using random-forest-based imitation learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2086–2093, 2019.

[2] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," *Robotics and Autonomous Systems*, vol. 112, pp. 72–83, 2019.

[3] D. Tanaka, S. Arnold, and K. Yamazaki, "Emd net: An encode–manipulate–decode network for cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1771–1778, 2018.

[4] S. Tokumoto and S. Hirai, "Deformation control of rheological food dough using a forming process model," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1457–1464.

[5] N. Koganti, T. Tamei, K. Ikeda, and T. Shibata, "Bayesian nonparametric learning of cloth models for real-time state estimation," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 916–931, 2017.

[6] J. H. Palep, "Robotic assisted minimally invasive surgery," *Journal of minimal access surgery*, vol. 5, no. 1, p. 1, 2009.

[7] B. Protyniak, J. Jorden, and R. Farmer, "Multiquadrant robotic colorectal surgery: the da vinci xi vs si comparison," *Journal of robotic surgery*, vol. 12, no. 1, pp. 67–74, 2018.

[8] P. M. Scheikl, E. Tagliabue, B. Gyenes, M. Wagner, D. Dall'Alba, P. Fiorini, and F. Mathis-Ullrich, "Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 560–567, 2022.

[9] H. Saeidi, J. D. Opfermann, M. Kam, S. Wei, S. Léonard, M. H. Hsieh, J. U. Kang, and A. Krieger, "Autonomous robotic laparoscopic surgery for intestinal anastomosis," *Science Robotics*, vol. 7, no. 62, p. eabj2908, 2022.

[10] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[11] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

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

[13] T. Collins, D. Pizarro, S. Gasparini, N. Bourdel, P. Chauvet, M. Canis, L. Calvet, and A. Bartoli, "Augmented reality guided laparoscopic surgery of the uterus," *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 371–380, 2020.

[14] P. Boonvisut and M. C. Çavuşoğlu, "Estimation of soft tissue mechanical parameters from robotic manipulation data," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 5, pp. 1602–1611, 2012.

[15] B. Frank, R. Schmedding, C. Stachniss, M. Teschner, and W. Burgard, "Learning the elasticity parameters of deformable objects with a manipulation robot," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1877–1883.

[16] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation," *arXiv preprint arXiv:2003.09044*, 2020.

[17] P. Zhou, J. Zhu, S. Huo, and D. Navarro-Alarcon, "Lasesom: A latent and semantic representation framework for soft object manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5381–5388, 2021.

[18] Z. Hu, P. Sun, and J. Pan, "Three-dimensional deformable object manipulation using fast online gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 979–986, 2018.

[19] B. Jia, Z. Hu, J. Pan, and D. Manocha, "Manipulating highly deformable materials using a visual feedback dictionary," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 239–246.

[20] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2017.

[21] L. Twardon and H. Ritter, "Learning to put on a knit cap in a head-centric policy space," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 764–771, 2018.

[22] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-d deformable object manipulation using deep neural networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4255–4261, 2019.

[23] 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.

[24] D. McConachie and D. Berenson, "Estimating model utility for deformable object manipulation using multiarmed bandit methods," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 967–979, 2018.

[25] B. Yang, B. Lu, W. Chen, F. Zhong, and Y.-H. Liu, "Model-free 3-d shape control of deformable objects using novel features based on modal analysis," *IEEE Transactions on Robotics*, 2023.

[26] A. Colomé and C. Torras, "Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 602–615, 2018.

[27] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1405–1412.

[28] E. Pignat and S. Calinon, "Learning adaptive dressing assistance from human demonstration," *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.

[29] B. Thach, B. Y. Cho, A. Kuntz, and T. Hermans, "Learning visual shape control of novel 3d deformable objects from partial-view point clouds," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8274–8281.

[30] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.

[31] T.-P. Fries, H. Matthies *et al.*, "Classification and overview of meshfree methods," Technische Universität Braunschweig, Tech. Rep., 2004.

[32] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[33] A. H. Barr, "Global and local deformations of solid primitives," in *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '84. Association for Computing Machinery, 1984, p. 21–30.

[34] M. Ristic and D. Brujic, "Efficient registration of nurbs geometry," *Image and vision computing*, vol. 15, no. 12, pp. 925–935, 1997.

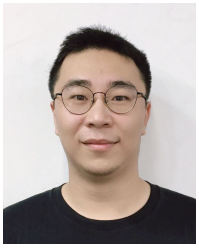
[35] R. Kolluri, "Provably good moving least squares," *ACM Transactions on Algorithms (TALG)*, vol. 4, no. 2, pp. 1–25, 2008.

[36] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A Multi-Model Framework for Interactive Physical Simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, ser. Studies in Mechanobiology, Tissue Engineering and Biomaterials, Y. Payan, Ed. Springer, Jun. 2012, vol. 11, pp. 283–321. [Online]. Available: https://hal.inria.fr/hal-00681539

[37] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[38] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[39] N. Marahrens, B. Scaglioni, D. Jones, R. Prasad, C. S. Biyani, and P. Valdastri, "Towards autonomous robotic minimally invasive ultrasound scanning and vessel reconstruction on non-planar surfaces," *Frontiers in Robotics and AI*, p. 178, 2022.
[40] Y. He, B. Liang, J. Yang, S. Li, and J. He, "An iterative closest points algorithm for registration of 3d laser scanner point clouds with geometric features," *Sensors*, vol. 17, no. 8, p. 1862, 2017.



Junlei Hu received the B.Eng. and M.Eng. degree in Mechanical Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017 and 2020. He is currently working toward the Ph.D. degree in medical robotics with the University of Leeds, Leeds, U.K.

His research interests include surgical robot, robotic planning, and computer vision.

Dominic Jones (Member, IEEE) received the M.Eng degree in Mechanical Engineering, the Ph.D. degree in Medical Mechatronics in 2015 and 2019, respectively.

He is currently a Lecturer (Assistant Professor) at the School of Electronic and Electrical Engineering. University of Leeds, Leeds, U.K. His research interests include autonomy in surgical robotics, tactile sensing in surgical tools, and skill assessment in surgical training.

Mehmet R. Dogar received the Ph.D. degree from the Robotics Institute, Carnegie Mellon University, in 2013. He is currently an Associate Professor at the School of Computing, University of Leeds, Leeds, U.K. Before joining Leeds, he was a Post-Doctoral Researcher at the Massachusetts Institute of Technology.

His research focuses on robotic planning. He is the Co-Chair of the IEEE-RAS Technical Committee on Mobile Manipulation and an Associate Editor for the IEEE Robotics and Automation Letters.

Pietro Valdastri (Fellow, IEEE) received the master's (Hons.) degree in Electronic Engineering from the University of Pisa, Pisa, Italy, in 2002, and the Ph.D. degree in Biomedical Engineering with Scuola Superiore Sant' Anna, Pisa, in 2006.

He is currently a Professor and the Chair in robotics and autonomous systems with the University of Leeds, Leeds, U.K. His research interests include robotic surgery, robotic endoscopy, design of magnetic mechanisms, and medical capsule robots.

Dr. Valdastri is a recipient of the Wolfson Research Merit Award from the Royal Society, a Consolidator Grant from the European Research Council, and a Career Award from the National Science Foundation.