Citation: I. Cuiral-Zueco, G. López-Nicolás. Multi-scale Laplacian-based FMM for shape control. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021), pp. 3792-3797. Prague, Czech Republic, September 27 - October 1, 2021.

Multi-scale Laplacian-based FMM for shape control

Ignacio Cuiral-Zueco and Gonzalo López-Nicolás

***Abstract*— Shape control has become a prominent research field as it enables the automation of tasks in many applications. Overall, deforming an object to a desired target shape by using few grippers is a major challenge. The limited information about the object dynamics, the need to combine small and large deformations in order to achieve certain target shapes and the non-linear nature of most deformable objects are factors that significantly hamper shape control performance. In this paper, we propose a shape control method for multi-robot manipulation of large-strain deformable objects. Our approach is based on multi-scale Laplacian descriptors that feed an FMM (Fast Marching Method) for elastic shape contour matching. The FMM's resulting path and the Laplacian operator are used to define a control strategy for the robot grippers. Simulation experiments carried out with an ARAP (As Rigid As Possible) deformation model provide satisfactory results.**

I. INTRODUCTION

Producing desired deformations in objects with the use of robotic manipulators is a challenge that can bring many advantages for automated industrial manufacturing, medical and domestic applications, among others. There is a large variety of deformable objects which can be classified into categories in terms of a broad range of characteristics. Similarly, the deforming actions performed on objects and the control strategies used for that purpose can be categorised using several criteria. Surveys such as [1] and [2] help with formalising an already wide-ranging problem. In [1], a deformable object classification regarding robot manipulation is proposed. Using a mixed criteria (physical and shape based), objects are classified into cloth-like, linear, planar and solid objects. The tasks that can be performed on each type of objects are also categorised, namely: tying knots (for linear objects), folding (for cloth-like objects and planar objects), hanging, splitting, cutting, etc. Our proposal focuses on the shape control task which, according to [1], concerns the large-strain object group (i.e. objects with a low Young's modulus). As for shape control, proposals vary significantly depending on the object type. However, they can be generally classified with respect to several common aspects. With regard to the deformation model, proposals can be model-based (discrete, continuous, learned, etc) or model-free. Another important general classification refers to the scale at which deformations occur, ranging from small and local to large and global deformations. The shape state estimation can be discrete or continuous (parametric) and the shape estimation process can be performed with vision sensors, force sensors or the combination of both.

I. Cuiral-Zueco and G. López-Nicolás are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. ignaciocuiral@unizar.es, gonlopez@unizar.es.

This work was supported by the Spanish Government/European Union through projects PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE) and COMMANDIA SOE2/P1/F0638 (Interreg Sudoe Programme, ERDF).

A. Related work

Proposals like [3] achieve local deformation shape control using several manipulators on planar objects. They use a discrete network of mass-spring-damper system along with a curve parametrisation in order to model the object and a dynamic energy-based control law. They achieve satisfactory simulation results and, since the error is only defined for the actuated points, a better convergence to the desired shape is accomplished as the number of available grippers increases. Combining shape control with deformable object transport, [4] introduces a consensus-based deformation model applied to broad flexible objects that require a large number of grippers in order to be manipulated. Tackling large isometric deformations on planar objects, [5] uses a Shape-from-Template (SfT) based algorithm and achieves proper performance on real experiments with monocular-based perception. Navarro-Alarcon et al. present diverse approaches such as [6] and [7]. In [6] the physical properties of the object model are presumed to be unknown and, thus, an adaptive deformation model is proposed. The method performs satisfactorily with 3D objects in real experiments. Another interesting proposal is [7], where the 2D object shape representation is based on a truncated Fourier series. The approach estimates deformation parameters to approximate the deformation Jacobian matrix. Real and simulated experiments with two grippers (one active and one passive) validate the method. Zhu et al. propose a shape control strategy based on an interaction matrix estimation by applying PCA (Principal Component Analysis) to the shape's 2D contour points [8]. The method is validated with simulation and real experiments using one active gripper and a passive element of the scene. In [9] they introduce, and validate with real experiments, a dual-arm flexible cable manipulation method that makes use of a Fourier series parametrisation. Hu et al. introduce in [10] a 3D deformable object servo control based on a GPR (Gaussian Process Regression) online learned object model. The method in [10] is overcome in [11], which combines FPFH (Fast Point Feature Histograms) and PCA in order to encode the state of the object. They achieve proper experimental results using a DNN-based (Deep Neural Networks) data-driven controller.

B. Shape control strategy proposal

Our proposal tackles the shape control problem for large-strain deformable objects with the use of a reduced number of grippers. We introduce multi-scale Laplacian surfaces combined with FMM (Fast Marching Method) for texture-less

deformable object contour matching. This formulation allows to perform a non-homogeneous elastic matching, which is an important aspect for accurately relating two deformation states. The multi-scale analysis allows prioritising the main dimensions of the object and still discriminate, given evenness in larger dimensions, the differences in smaller regions. In contrast to classic elastic 1D curvature matching, Laplacian coordinates generalise the average curvature of the contour points in three dimensions. Provided a Laplacian-based matching between current and target shapes, we design a control law based on the minimisation of a linearly precise smoothness energy function. This energy function is defined using the previously computed point-matching information. We do not rely on texture and generalise contour analysis to three dimensions. The method offers a model-free approach with no dependence on learning or parameter initialisation. Furthermore, although seeking applications with a reasonably reduced number of robotic grippers, this number is not restricted within the method. We perform satisfactory simulation tests using the As Rigid As Possible (ARAP) deformation model [12]. In the tests, an RGB-D camera sensor is also simulated to make the object surface data points resemble more realistic and noisy conditions.

II. PROBLEM FORMULATION

Shape control solves the problem of bringing a deformable object to a deformation state so that it achieves a desired shape. In this proposal we focus on large-strain objects, which tend to behave poorly when manipulated with a large number of grippers due to their high local rigidity. Seeking generality, we consider the object to be texture-less so fiducial markers or patterns are not required. The input information is assumed to be provided by a depth sensing device such as an RGB-D camera.

A. Problem setup

The object's visible surface can be segmented as a 3D surface Θ (see Fig. 1), perceived at the sensor's resolution. In our problem formulation we assume that $\gamma_g \in \Gamma = \{\gamma_g, g = 1, ..., \mathrm{G}\}$ grippers can be placed around the contour $\partial\Theta$ of Θ. The goal is to define a series of control actions such that $\partial\Theta$ resembles the target object contour $\partial\bar{\Theta}$. Since we do not rely on texture information, our method is based on the discretised contour points of $\partial\Theta$ and $\partial\bar{\Theta}$. The discrete current and target shape contour points are $v_n \in V = \{v_n, n = 1, \ldots, N\}$ and $\bar{v}_p \in \bar{V} = \{\bar{v}_p, p = 1, \ldots, P\}$ respectively. Each current and target contour point defines a local reference $\mathbf{T}_{nw}, \bar{\mathbf{T}}_{pw} \in \mathbb{R}^{4\times4}$ with respect to the world global reference frame. The local references axis of each point v_n are defined as $(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n)$, with $\mathbf{x}_n$ the contour's tangent vector at point v_n; $\mathbf{y}_n$ the locally normal surface vector at v_n; and $\mathbf{z}_n$ orthonormal to $\mathbf{x}_n$ and $\mathbf{y}_n$.

The current contour nodes define graphs $\mathcal{G}^\lambda = (V, E^\lambda)$ with vertices, $v_n \in V$. Graphs' edges $E^\lambda = \{e^\lambda_m,\ m = 1, \ldots, M^\lambda\}$ link nodes V lying within a topological neighbouring scale $s^\lambda \in S = \{s^\lambda, \lambda = 1, \ldots, \Lambda\}$, where s^λ defines the neighbouring radius along $\partial\Theta$. Therefore, for each value of λ, a new graph with the same nodes V, but with a different set of edges E^λ, is obtained. Analogously, target contour points define graphs $\bar{\mathcal{G}}^\lambda = (\bar{V}, \bar{E}^\lambda)$ with vertices, $\bar{v}_p \in \bar{V}$ and edges $\bar{E}^\lambda = \{\bar{e}^\lambda_q,\ q = 1, \ldots, Q^\lambda\}$.

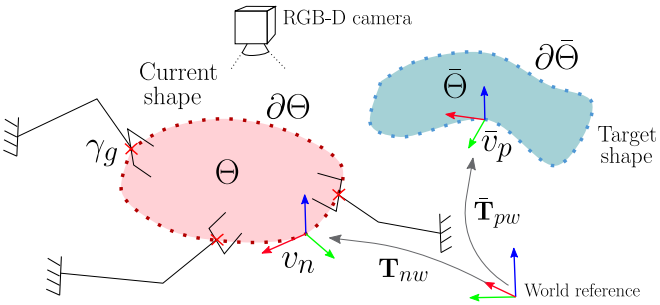

Fig. 1: Problem set-up. Current and target shape contour points $(v_n, \bar{v}_p)$ extracted from contours $\partial\Theta$ and $\partial\bar{\Theta}$ respectively. The object is grabbed by G grippers at γ_g points. Affine transforms $\mathbf{T}_{nw}$ and $\bar{\mathbf{T}}_{pw}$ define each of the contours discrete points' v_n and $\bar{v}_p$ local references.

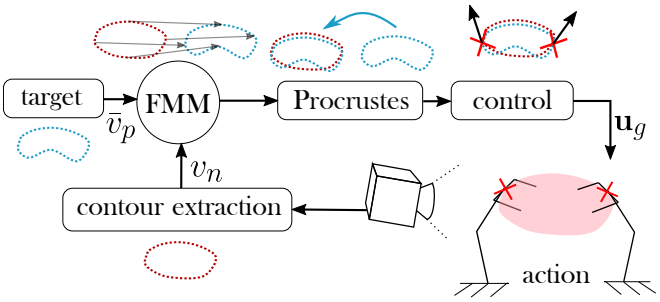

Fig. 2: Control scheme. For each iteration current contour points v_n are extracted from the RGB-D information. An FMM Laplacian-based elastic matching is performed between target contour points $\bar{v}_p$ and v_n. Matched target points $\bar{v}_n$ undergo a rigid Procrustes transform in order to minimise matching nodes distances. Using the matching information, the control strategy generates an action $\mathbf{u}_g$ for the grippers.

B. Method's general overview

The general control scheme is represented in Fig. 2. The current contour points v_n may be obtained from RGB-D images using an α-shape contour extraction method. We perform an FMM-based elastic contour matching between V and $\bar{V}$. Using the contour matches, a Procrustes transform is applied to the target shape in order to minimise the rigid transform error between current contour points v_n and their matched target points $\bar{v}_n$ [13]. We define our control strategy using the contour matches and the transforms $\mathbf{T}_{nw}$ and $\bar{\mathbf{T}}_{nw}$. Being $\bar{\mathbf{T}}_{nw}$ the transforms of the Procrustes-updated matched target points $\bar{v}_n$. A single integrator model defines gripper dynamics as $\dot{\gamma}_g = \mathbf{u}_g = (u_{g,x}, u_{g,y}, u_{g,z})$, being $\dot{\gamma}_g$ the gripper's velocity and $\mathbf{u}_g \in \mathbb{R}^3$ its control input.

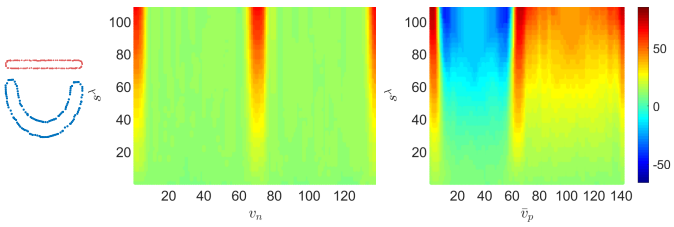

Fig. 3: Laplacian surfaces Ω_y and $\bar{\Omega}_y$ for neighbouring scales s^λ. Ω_y corresponds to a bar-shaped (red) contour and $\bar{\Omega}_y$ to a "U" shaped contour (blue).

III. Laplacian-based FMM for contour matching

Making use of graphs $\mathcal{G}^\lambda$ and $\bar{\mathcal{G}}^\lambda$ we propose a set of multi-scale Laplacian surfaces. These surfaces are used in a Fast Marching Method (FMM) optimisation in order to obtain an elastic contour matching between v_n and $\bar{v}_p$. As a result, we obtain a new set of target contour points $\bar{V}_{match} = \{\bar{v}_n,\ n = 1, \ldots, N\}$ that become a match for each of the current contour points v_n.

A. Multi-scale Laplacian surfaces

The random walk Laplacian operator for a given graph $\mathcal{G}$ is defined by the positive semi-definite matrix:

$$\mathcal{L} = \mathbf{I}_N - \mathbf{D}^{-1}\mathbf{A}, \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{N\times N}$ is the node degree diagonal matrix and $\mathbf{A} \in \mathbb{R}^{N\times N}$ the adjacency matrix of graph $\mathcal{G}$. Matrix $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is the identity matrix. This Laplacian operator is computed for each of the current and target contour graphs $\mathcal{G}^\lambda$ and $\bar{\mathcal{G}}^\lambda$ at every scale λ obtaining: $\mathcal{L}^\lambda \in \mathbb{R}^{N\times N}$ and $\bar{\mathcal{L}}^\lambda \in \mathbb{R}^{P\times P}$. Multiplying the random walk Laplacian at scale λ by the v_n nodes we obtain a set of N Laplacian coordinate vectors $\mathbf{l}_n^\lambda$:

$$\mathbf{L}^\lambda = \mathcal{L}^\lambda \mathbf{V}, \tag{2}$$

with $\mathbf{L}^\lambda \in \mathbb{R}^{N\times 3}$. Matrix $\mathbf{V} \in \mathbb{R}^{N\times 3}$ is obtained by stacking the v_n nodes. Using $\mathbf{T}_{nw}$ we obtain the Laplacian vectors in local reference, with local components: $\mathbf{l}_{n,x}^\lambda$, $\mathbf{l}_{n,y}^\lambda$ and $\mathbf{l}_{n,z}^\lambda$. We define a Laplacian discrete surface $\Omega \in \mathbb{R}^{N\times\Lambda}$ across scales s^λ for a given local component x as $\Omega_x = \left[\mathbf{l}_{n,x}^\lambda\right]$ with columns $\lambda = 1, \ldots, \Lambda$ and rows $n = 1, \ldots, N$. For the current shape contour we compute Ω_x, Ω_y and Ω_z. Analogously, the local coordinates of the target contour Laplacian vectors $\bar{\mathbf{l}}_p^\lambda$ are obtained: $\bar{\mathbf{l}}_{p,x}^\lambda$, $\bar{\mathbf{l}}_{p,y}^\lambda$ and $\bar{\mathbf{l}}_{p,z}^\lambda$ along with the target Laplacian surfaces $\bar{\Omega} \in \mathbb{R}^{P\times\Lambda}$, obtaining $\bar{\Omega}_x, \bar{\Omega}_y$ and $\bar{\Omega}_z$. In Fig. 3, two Laplacian surfaces Ω_y and $\bar{\Omega}_y$ can be visualised.

B. Laplacian-based FMM for contour matching

The Fast Marching Method (FMM) [14] solves the Eikonal equation

$$\left|\nabla T(\theta,\bar{\theta})\right| F(\theta,\bar{\theta}) = 1, \quad \theta \in \partial\Theta,\ \bar{\theta} \in \partial\bar{\Theta}, \tag{3}$$

which is typically used to model the propagation of a surface front moving with a normal speed $F(\theta,\bar{\theta})$ and crossing a point $(\theta,\bar{\theta})$ with a time cost of $T(\theta,\bar{\theta})$. $T(\theta,\bar{\theta})$ can be seen as a cost function in which the cost of passing through point $(\theta,\bar{\theta})$ translates in more travel time for higher $T(\theta,\bar{\theta})$ values. The FMM is widely used nowadays in diverse optimisation problems and computer vision applications as it allows solving the fastest route of a propagation front and thus, finding the shortest path given a speed function. Contour matching can be performed through FMM by solving the front propagation along the similarity surface of two contours. FMM has been used to perform curvature-based contour matching between discrete curves with sub-resolution accuracy [15]. We propose an FMM contour matching approach generalising curvature to three dimensions and multi-scale analysis by means of Laplacian surfaces Ω as our descriptors.

We define a discrete surface $\mathbf{F}_x \in \mathbb{R}^{N\times P}$ as similarity matrix for a given local component x:

$$\mathbf{F}_x(n,p) = \sum_{\lambda=1}^{\Lambda} \left(\left|\bar{\Omega}_x(p,\lambda) - \Omega_x(n,\lambda)\right| + \beta\right)^{-1}, \tag{4}$$

where $\beta > 0$, which implies $\mathbf{F}_x(n,p) > 0$. Once $\mathbf{F}_x$, and similarly $\mathbf{F}_y$ and $\mathbf{F}_z$, are obtained we compute $\mathbf{F}$ as:

$$\mathbf{F}(n,p) = \left\|(\mathbf{F}_x(n,p), \mathbf{F}_y(n,p), \mathbf{F}_z(n,p))\right\|_2. \tag{5}$$

With $\mathbf{F}(n,p) > 0$. Note that $\mathbf{F}$ defines a discrete speed function surface in which row indices represent current contour points and column indices represent target contour points. Thus, values of $\mathbf{F}$ correlate current and target local Laplacian vector components. We perform the Multi Stencil Fast Marching Method [16] with $\mathbf{F}$ as our input speed surface and obtain the discrete cost function $\mathbf{T}$. The gradient descent path $\mathbf{P}$ from $\mathbf{T}(N,P)$ to $\mathbf{T}(1,1)$ defines the matching (Fig. 4). Interpolating the current contour points $v_n \in V$ in path $\mathbf{P}$, the target contour matched nodes $\bar{v}_n \in \bar{V}_{match}$ are obtained.

Note that $\partial\Theta$ and $\partial\bar{\Theta}$ are closed contours and thus an initial matching point must be defined. The initial matching point is obtained, as suggested in [15], by stacking $\mathbf{F}$ twice on each dimension: $\hat{\mathbf{F}} = \left[(1,1)^T, (1,1)^T\right] \otimes \mathbf{F}$, with $\otimes$ the Kronecker product. $\hat{\mathbf{F}}$ is used to generate $\hat{\mathbf{T}}$, in which the gradient descent's end-up point converges to the initial match. Once the initial match is obtained, $\mathbf{F}$ is shifted accordingly to bring the initial match coordinates to the first element $\mathbf{F}(1,1)$ and the descent path $\mathbf{P}$ along $\mathbf{T}$ is recomputed.

In order to minimise the distance between the set of current nodes V and the set of target nodes $\bar{V}$, a Procrustes rigid transform [13] is performed before generating the control action. The inputs for Procrustes are the current contour set of nodes V and their corresponding nodes on the target contour $\bar{V}_{match}$.

IV. Control strategy definition

Our proposal for the control strategy is based on a generic quadratic mesh deformation energy function. This kind of function is used in the field of computer graphics for solving high resolution mesh deformations with the position change of a few handle points as an input. For this purpose, works such as [17] propose methods to obtain deformation subspaces that allow obtaining a linearly precise point mapping

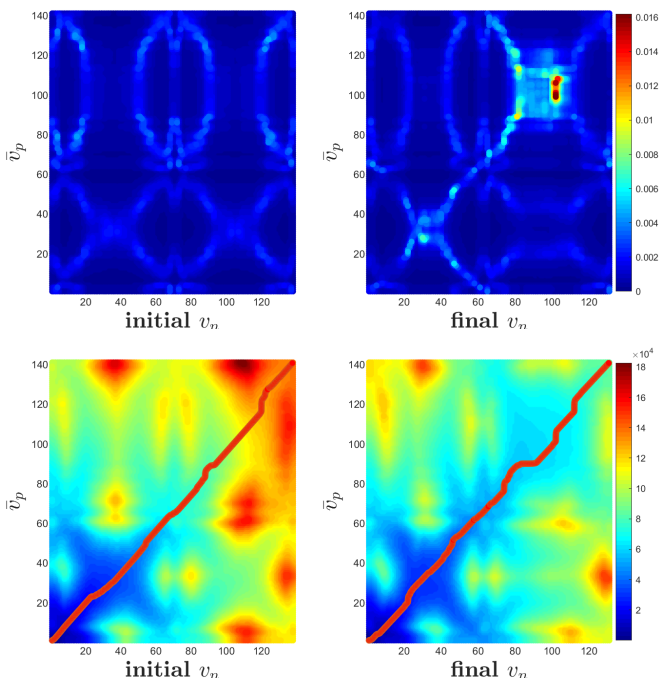

Fig. 4: Visualisation of matrices **F**. Below them, their corresponding **T** matrix is represented with the gradient descent path **P** (red line). On the left, **F** compares the Laplacian surfaces Ω and $\bar{\Omega}$ whose y components are shown in Fig. 3. On the right, the comparison of a more advanced state of the shape control process in which a bar-shaped object begins to acquire a "U" shape. This translates in larger values along the **F**'s diagonal and a more prominent valley in **T**.

that preserves local features. Our proposal makes use of their formulation but with an opposite perspective. Instead of solving the mapping of numerous surface points as a function of a few handles' position, we seek to solve the position of a few grippers already knowing the mapping correspondences (i.e. the matching) of a considerable amount of surface contour points. Furthermore, using a linearly precise smoothness energy suits our goal of manipulating large-strain and locally rigid objects where pronounced local deformations may be problematic. We define our optimisation variable $\mathbf{X} \in \mathbb{R}^{(N+G)\times 3}$, which encloses the target positions of the current contour nodes V and the grippers Γ. Using $\mathbf{X}$, a quadratic energy optimisation function is defined for obtaining a solution denoted by $\bar{\mathbf{X}} \in \mathbb{R}^{(N+G)\times 3}$:

$$\bar{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2}\text{trace}\,(\mathbf{X}^\intercal \mathbf{B}\mathbf{X}) \quad s.t. \quad \mathbf{C}\mathbf{X} = \mathbf{J}\mathbf{H}. \tag{6}$$

Where $\mathbf{C} \in \mathbb{R}^{N\times(N+G)}$ retrieves the contour points target positions from the optimisation variable $\mathbf{X}$. Matrix $\mathbf{J} \in \mathbb{R}^{N\times 4N}$ stores the position of the current contour nodes V in homogeneous coordinates. Matrix $\mathbf{H} \in \mathbb{R}^{4N\times 3}$ stores $([\mathbf{I}_3, \mathbf{0}]\mathbf{T}_n)^\intercal$, where affine transforms $\mathbf{T}_n \in \mathbb{R}^{4\times 4}$ are obtained as $\mathbf{T}_n = (\bar{\mathbf{T}}_{nw})^{-1}\mathbf{T}_{nw}$. These transforms, $\mathbf{T}_n$, bring the current contour nodes' V local references $\mathbf{T}_{nw}$ to their corresponding nodes' $\bar{V}_{match}$ local references $\bar{\mathbf{T}}_{nw}$.

The constraint in (6) states that the matched points V are mapped to their homologous $\bar{V}_{match}$. **B** is the linearly precise smoothness energy:

$$\mathbf{B} = \mathcal{L}^\intercal \mathbf{M}^{-1}\mathcal{L}. \tag{7}$$

We approximate the mass matrix $\mathbf{M} \in \mathbb{R}^{(N+G)\times(N+G)}$ as a matrix that stores on its diagonal the average distance between a contour node v_n and its neighbours. In (7), both the Laplacian operator $\mathcal{L}$ and the mass matrix **M** are defined according to a 2-closest-neighbours topological search within the contour $\partial\Theta$ (this implies two neighbours per node). For solving (6) we use the linearization proposed in [17]:

$$\bar{\mathbf{X}} = (\mathbf{C}^\intercal\mathbf{J} - \bar{\mathbf{C}}^\intercal(\bar{\mathbf{C}}\mathbf{B}\bar{\mathbf{C}}^\intercal)^{-1}\bar{\mathbf{C}}\mathbf{B}\mathbf{C}^\intercal\mathbf{J})\mathbf{H}, \tag{8}$$

where $\bar{\mathbf{C}} \in \mathbb{R}^{G\times(N+G)}$, in the fashion of **C** for contour points, acts as a gripper point selector on $\bar{\mathbf{X}}$. Once $\bar{\mathbf{X}}$ is obtained, we define the control:

$$\mathbf{U} = k(\bar{\mathbf{C}}\bar{\mathbf{X}} - \mathbf{\Gamma}). \tag{9}$$

Matrices **U** and $\mathbf{\Gamma}$ are obtained by stacking $\mathbf{u}_g$ gripper actions and $\boldsymbol{\gamma}_g$ gripper positions respectively. Parameter $k > 0$ is the proportional control gain.

A. Stability Analysis

Equations (6) and (7) define the minimisation of a discrete approximation to a squared Laplacian integral along the contour $\partial\Theta$. Using **B** as our energy is the discrete equivalent to solving the continuous bi-harmonic equation $\Delta^2 u = 0$, where $\Delta^2 = \nabla^2\nabla^2$ and ∇^2 is the square continuous Laplacian operator. Given the uniform spatial distribution of the contour points and the topological neighbouring on $\partial\Theta$, we assume the graph Laplacian $\mathcal{L}$ to converge to his continuous homologous. This allows us to analyse our formulation as the solution of the smooth Dirichlet problem which, in Lipschitz domains, is proven to have solution uniqueness [18]. Our formulation defines $D_n \subset \partial\Theta$ concatenated Lipschitz domains and thus points (i.e. grippers γ_g) falling within domains D_n present uniqueness of solution in $\bar{\mathbf{X}}$. Conditions in (6) imply Dirichlet conditions on disjoint regions of $\partial\Theta$ for the bi-harmonic equation. Note that, for each gripper γ_g, Dirichlet conditions are defined by its surrounding nodes affine transforms. Considering solution uniqueness, we can assume $\bar{\mathbf{C}}\bar{\mathbf{X}}$ to be locally constant and define the Lyapunov function:

$$\mathcal{V} = \sum_{g=1}^{\mathrm{G}} \mathcal{V}_g, \text{ with } \mathcal{V}_g = \frac{1}{2}(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g)(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g)^\intercal. \tag{10}$$

Matrices sub-index g refers to the vector conformed by their g-th row. In (10), $\mathcal{V}_g(\mathbf{0}) = \mathbf{0}$ for $(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g) = \mathbf{0}$ and $\mathcal{V}_g > \mathbf{0} \;\; \forall(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g) \neq \mathbf{0}$. Deriving (10) and substituting (9) we obtain:

$$\dot{\mathcal{V}}_g = -k(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g)(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g)^\intercal. \tag{11}$$

Which ensures $\dot{\mathcal{V}}_g \leq \mathbf{0} \;\; \forall(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g)$ and $\dot{\mathcal{V}}_g < \mathbf{0} \;\; \forall(\bar{\mathbf{C}}_g\bar{\mathbf{X}} - \mathbf{\Gamma}_g) \neq \mathbf{0}$ and thus local asymptotic stability in the sense of Lyapunov.

V. EXPERIMENTAL RESULTS IN SIMULATION

We have performed several simulations using the As Rigid As Possible (ARAP) deformation model as it fits our purpose of simulating large-strain objects [12]. Figure 5 (and video attachment) show seven experiments of different characteristics. On the first column: the initial state of the deformable object (red triangulation) and the gripper positions γ_g (black dots with red crosses overlapped) are displayed. The current and the target contours (v_n and $\bar{v}_p$) (obtained with a simulated RGB-D camera) are displayed as well in red and blue respectively. Thin grey lines between the contours represent the matching obtained by using the proposed Laplacian based Fast Marching Method. The second column shows the same elements after the presented shape control process converges. Third column displays the relative boundary error distribution on each iteration. The boundary error for a node v_n is computed as the Euclidean distance between node v_n and its assigned target contour node $\bar{v}_n$. On the fourth column, the relative object deformation distribution is represented as it varies along iterations. This deformation metric is obtained by computing the variation in length of the simulation mesh edges with respect to their initial length. All the tests have been performed with control gain $k = 0.2$ and the initial gripper positions are presumed to be known.

The first three experiments perform deformations on a more global scale. Note how experiments 1 (bending) and 3 (unfolding) are opposites and, although their boundary error convergence is similar, their object deformation distributions are complementary: predominant negative values (compression) for the bending process and positive values (distension) for the unfolding process. The second experiment has the same initial set up as the first one but one of the grippers (green square) is kept fixed. Although this affects the method's performance, it converges to an adequate solution. The remaining three experiments gradually represent cases of deformation on more local scales. In experiment 4 (triangles), the rigid nature of ARAP becomes clear and the shape control method struggles at reducing the boundary error. Note how most edges suffer large compression on the deformation graph. In opposition to experiment 4, experiment 5 (star) shows large distension on edges. The boundary error is reduced yet several points fall far away the error distribution's median. These points correspond to the star tips which, in the absence of grippers, cannot be controlled. The sixth experiment (bone) shows an isolated local deformation. This is reflected on the boundary error at the first iteration: the boundary error distribution groups around a low median value, but several contour points lie well apart. Similarly, the object's deformation values are generally low. Just a few locally affected points suffer compression or distension. Last row in Figure 5 shows a 3D deformation experiment in which a bar is bent along x and z axis. For ease of visualisation, the simulation mesh colour map represents the z coordinate values. Note on its relative deformation graph how several edges undergo large deformations during the shape control process but approach their initial values on the final shape.

VI. CONCLUSIONS

We have developed a shape control method for large-strain deformable object manipulation with few grippers. The method involves an elastic contour matching performed with a novel multi-scale Laplacian surface descriptor combined with FMM optimisation. We proposed a control strategy based on the minimisation of the Laplacian deformation energy. Local convergence analysis and simulations validate our proposal. The non-reliance on object texture, a three-dimensional analysis, the lack of dependence on both model and parameters and the absence of a restricted number of grippers are some of the advantages of our proposal.

REFERENCES

[1] 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*, 37(7):688–716, 2018.

[2] R. Herguedas, G. López-Nicolás, R. Aragüés, and C. Sagüés. Survey on multi-robot manipulation of deformable objects. In *24th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 977–984, 2019.

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

[4] G. López-Nicolás, R. Herguedas, M. Aranda, and Y. Mezouar. Simultaneous shape control and transport with multiple robots. In *IEEE Int. Conf. on Robotic Computing*, pages 218–225, 2020.

[5] M. Aranda, J.A. Corrales Ramon, Y. Mezouar, A. Bartoli, and E. Özgür. Monocular visual shape tracking and servoing for isometrically deforming objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7542–7549, 2020.

[6] 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*, 32(2):429–441, 2016.

[7] 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*, 34(1):272–279, 2017.

[8] J. Zhu, D. Navarro-Alarcon, R. Passama, and A. Cherubini. Vision-based manipulation of deformable and rigid objects using subspace projections of 2D contours. *Robotics and Autonomous Systems*, 142:103798, 2021.

[9] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini. Dual-arm robotic manipulation of flexible cables. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 479–484, 2018.

[10] Z. Hu, P. Sun, and J. Pan. Three-dimensional deformable object manipulation using fast online Gaussian process regression. *IEEE Robotics and Automation Letters*, 3(2):979–986, 2018.

[11] 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*, 4(4):4255–4261, 2019.

[12] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[13] J. C. Gower and G. B. Dijksterhuis. *Procrustes problems*. Oxford University Press, 2004.

[14] J. A Sethian and A.M. Popovici. 3-D traveltime computation using the fast marching method. *Geophysics*, 64(2):516–523, 1999.

[15] M. Frenkel and R. Basri. Curve matching using the fast marching method. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 35–51, 2003.

[16] M. Sabry Hassouna and A.A. Farag. Multistencils fast marching methods: A highly accurate solution to the Eikonal equation on Cartesian domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1563–1574, 2007.

[17] Y. Wang, A. Jacobson, J. Barbič, and L. Kavan. Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics*, 34(4):1–11, 2015.

[18] C.E. Lawrence. *Partial Differential Equations*. American Mathematical Society, Providence, Rhode Island, 1998.

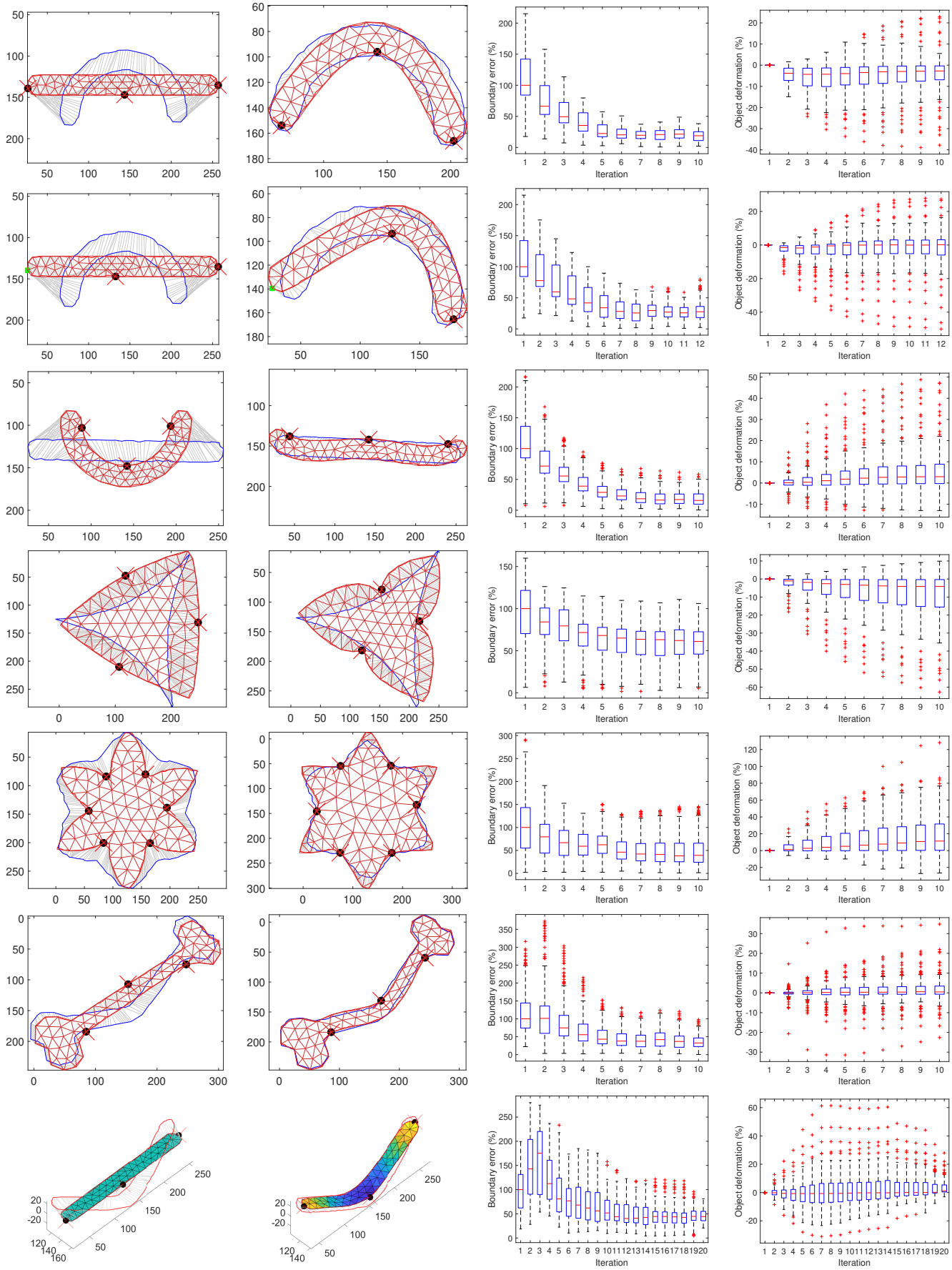

Fig. 5: Results of 7 simulations. See section V for a detailed explanation.