A Depth-Based Algorithm for Manipulating Deformable Objects Using Smooth Parametric Surfaces and Energy Minimisation

David Navarro-Alarcon, Omar Zahra

To cite this version:

David Navarro-Alarcon, Omar Zahra. A Depth-Based Algorithm for Manipulating Deformable Objects Using Smooth Parametric Surfaces and Energy Minimisation. IEEE International Conference on Robotics and Automation, May 2018, Brisbane, Australia. hal-01770418

HAL Id: hal-01770418
https://hal.archives-ouvertes.fr/hal-01770418

Submitted on 28 Apr 2018

A Depth-Based Algorithm for Manipulating Deformable Objects Using Smooth Parametric Surfaces and Energy Minimisation

David Navarro-Alarcon and Omar Zahra

***Abstract*— In this brief work, we present a new method for controlling deformations of soft objects by using parametric surfaces as a new type of deformation feedback features. This new approach allows us to actively deform objects into complex 3D shapes. A kinematic-based motion controller is derived using an energy minimisation strategy.**

I. THE METHOD

Consider a set of R robotic manipulators (or fingers) that rigidly grasp a deformable object O. We denote the 3D position of the ith robot (measured at the time instant t) with the vector $\mathbf{r}_t^i$. The total position vector of the R robots is denoted by $\mathbf{r}_t = [\mathbf{r}_t^1, \ldots, \mathbf{r}_t^R]^\intercal \in \mathbb{R}^{3R}$. To measure the object's deformations, we use a static RGB-D sensor (e.g. Kinect) that continuously observes the manipulated object. Let us denote by $\mathbf{p}$ a 3D point located over the portion of object's surface that can be observed by the image sensor. It is assumed that this surface is smooth and twice differentiable, therefore, its 3D coordinates can be parameterised as $\mathbf{p} = [x, y, f(x, y)]$. However, since the object's deformation model is unknown, the analytical structure of $z = f(x, y)$ is also not known.

The object's feedback surface can be approximated with the following parametric function:

$$
\begin{aligned}
f(x,y) = o + \sum_{j=0}^{M-1}\sum_{i=0}^{N-1} \begin{bmatrix} a^{i,j} & b^{i,j} & c^{i,j} & d^{i,j} \end{bmatrix} \cdot \\
\begin{bmatrix} \cos(ix\pi/l_x) & \sin(ix\pi/l_x) & \cos(jy\pi/l_y) & \cos(jy\pi/l_y) \end{bmatrix} \\
= \mathbf{g}(x,y) \cdot \mathbf{s} \quad (1)
\end{aligned}
$$

where $N, M > 0$ is the number of harmonics taken into account, $\mathbf{s} = [a^{i,j}, b^{i,j}, c^{i,j}, d^{i,j}, o]^\intercal$ represent the vector of constant surface parameters, and $l_x, l_y > 0$ are scaling factors. Note that the vectorial function $\mathbf{g}(x, y)$ is known and can be constructed with feedback data.

To approximate the unknown parameters $\mathbf{s}$, we first collect L data points of $\mathbf{p}^k = [x^k, y^k, z^k]$ at various locations over the surface, and then construct the long vector and matrix

$$\vec{\mathbf{z}} = \begin{bmatrix} z^1 & \cdots & z^L \end{bmatrix}^\intercal \quad (2)$$

$$\vec{\mathbf{G}} = \begin{bmatrix} \mathbf{g}(x^1, y^1)^\intercal & \cdots & \mathbf{g}(x^L, y^L)^\intercal \end{bmatrix}^\intercal \quad (3)$$

With this data, the vector $\mathbf{s}_t$ of surface parameters can be computed at every time instant t as follows:

$$\mathbf{s}_t = \left(\vec{\mathbf{G}}^\intercal \vec{\mathbf{G}}\right)^{-1} \vec{\mathbf{G}}^\intercal \vec{\mathbf{z}} \quad (4)$$

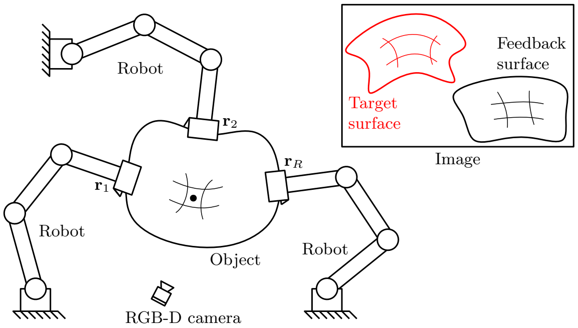

Fig. 1. The robotic system manipulating a deformable object.

The vector $\mathbf{s}_t$ can be used to characterise the object's feedback surface. To manipulate/deform the object into a target configuration, we must find the relation between the (input) motion of the robots and the (output) surface parameters [1], [2]. We approximate this unknown model with the following linearly parametrisable expression:

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{A}_t \mathbf{u}_t \quad (5)$$

where the mapping matrix $\mathbf{A}_t$ represents the deformation properties of the object and the vector $\mathbf{u}_t = \mathbf{r}_t - \mathbf{r}_{t-1}$ stands for the controllable motions of the robots. This unknown deformation matrix $\mathbf{A}_t$ can be computed online by using the method described in [3].

Given a desired surface $f_d(x, y)$ described by the parameters $\mathbf{s}^*$, a controller is derived by minimising the following quadratic cost function:

$$\mathcal{P} = \langle \mathbf{s}_{t+1} - \mathbf{s}^* , \mathbf{s}_{t+1} - \mathbf{s}^* \rangle_{\mathbf{H}} + \langle \mathbf{u}_t , \mathbf{u}_t \rangle_{\mathbf{Q}} \quad (6)$$

for $\mathbf{H} = \mathbf{H}^\intercal > 0$ and $\mathbf{Q} = \mathbf{Q}^\intercal > 0$ as weight matrices. Computing the extremum $\nabla \mathcal{P}(\mathbf{u}_t) = \mathbf{0}$ and solving for $\mathbf{u}_t$ yields the motion control input:

$$\mathbf{u}_t = -\left(\mathbf{A}_t^\intercal \mathbf{H} \mathbf{A}_t + \mathbf{Q}\right)^{-1} \mathbf{A}_t^\intercal \mathbf{H} \left(\mathbf{s}_t - \mathbf{s}^*\right) \quad (7)$$

that iteratively minimises $\mathcal{P}$. Note, however, that this method only guarantees local minimisation of the cost function.

REFERENCES

[1] D. Navarro-Alarcon and Y.-H. Liu, "Uncalibrated vision-based deformation control of compliant objects with online estimation of the Jacobian matrix," in *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, 2013, pp. 4977–4982.

[2] D. Navarro-Alarcon, H. Yip, Z. Wang, Y.-H. Liu, *et al.*, "Automatic 3D manipulation of soft objects by robotic arms with adaptive deformation model," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 429–441, 2016.

[3] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2D image shapes," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 272–1279, 2018.

This work is supported in part by the HKPolyU under grant 4-ZZHJ, and in part by the HK RGC under GRF grant 14203917.

The authors are with the Department of Mechanical Engineering, Hong Kong Polytechnic University, HKSAR. e-mail: `dna@ieee.org`