2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Direct TPS-based 3D non-rigid motion estimation on 3D colored point cloud in eye-in-hand configuration

Lénaïc Cuau[1], João Cavalcanti Santos[1], Philippe Poignet[1], and Nabil Zemiti[1]

***Abstract*— In this paper, a method for 3D non-rigid motion estimation of a surface using an RGB-D camera in eye-in-hand configuration is presented. The eye-in-hand configuration eliminates errors typically associated with camera-end-effector calibration, and is thus desirable for task on moving surfaces such as bioprinting. However, its implementation is challenging since camera and surface of interest are moving, making mesh-based approaches unsuitable. Thus, the proposed method operates directly on point clouds, benefiting from accurate and simplified data processing. A point cloud contains both intensity and depth data, with the former used to estimate in-plane deformation and the latter to compute full 3D deformation. Surface deformation is modeled via a Thin Plate Spline model. The method accuracy is assessed at 0.1 mm accuracy in simulated datasets, rendering it suitable for precision tasks, and its feasibility is validated experimentally on a moving platform that deforms at a rate of 0.8 Hz with a 4 mm in-plane amplitude and a 20 mm elevation amplitude.**

I. INTRODUCTION

Unlike rigid objects, the robotic interaction with non-rigid objects is a relatively recent scientific topic [1]. In this context, a typical objective involves following a path on a deforming surface, that is to say positioning the end-effector relative to a designated deforming target as illustrated in Fig. 1. This objective bears significance in various practical contexts such as additive manufacturing [2] or *in situ* bioprinting [3] which is the application aimed in this work. Traditionally, the path following task relies on controlling the tool's position with respect to the deforming surface using sensors feedback. For instance, as demonstrated in [2], a 1D laser sensor can be used for real-time control of tool-to-surface distance, even in the presence of surface motion. Nevertheless, these sensors solely provide measurement along a one-dimensional axis and are incapable of addressing non-rigid motions such as expansion, retraction, or in-plane displacement. Consequently, the integration of additional information becomes essential. Zhu *et al.* [3] utilized a 2D camera and markers placed on a deformable lung to directly print devices onto it. By linking marker positions with surface deformation through deep learning techniques, they managed to precisely print a specific shape on a deforming surface. However, in a medical context marker utilization can be cumbersome for patients and not always feasible. Therefore, this paper proposes relying solely on information provided by an RGB-D camera to recover surface 3D deformation. RGB-D cameras offer great possibilities in computer vision and 3D world understanding giving dense 3D colored point clouds instead of 2D images. The use of RGB-D cameras instead of multi-cameras system for non-rigid tracking seems promising given their compactness, depth noise reduction, and enhanced depth accuracy [4], [5].

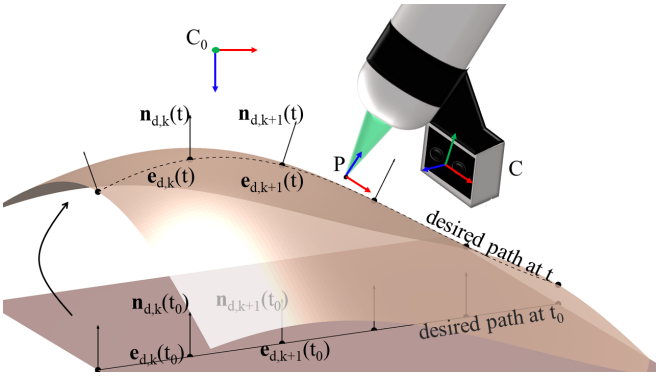

Fig. 1. Illustration of the considered problem: the end-effector P, guided through camera's observations, must follow a path represented by a sequence of desired positions $\mathbf{p}_{d,k}$ and orientations $\mathbf{n}_{d,k}$ on a moving surface. These vectors are function of time since the desired path should follow the surface displacements.

The positioning of this camera relative to the robot is essential for efficient servoing. Usually, the direct observation of the end-effector is desirable as it eliminates the necessity of relying on the established kinematic relationship between the end-effector and the camera. This approach is referred to as an eye-in-hand endpoint-closed loop (ECL) configuration [6]. However, it represents a challenging task in terms of image processing as the observed area is partial and surface of interest is moving, and deforming. Indeed, while executing the task, the camera's view is centered on the end-effector, resulting in only a partial changing view of the entire surface. The state-of-the-art tracking methods using RGB-D cameras are usually developed to track entirely visible objects that are manipulated under a static camera. So pre-constructed models such as Finite Element Model [5], [7] or geometric models [4], [8] are employed. These "pre-construction then tracking" approaches imply a computationally expensive construction of the model before tracking. In consequence, such methods are viable exclusively when observing an entire deforming object that remains consistently visible under static camera condition which is not the case in the ECL configuration. In scenarios where the

This work was funded by the French project ANR-21-ASM2-0001 – Bloc-Print2, the French Ministry of Defence - Defence Innovation Agency, and the Region Occitanie - "Défi-clé Robotique centrée sur l'humain"

[1] LIRMM, Univ. Montpellier, CNRS, Montpellier, France [lcuau, joao.cavalcanti-santos, philippe.poignet, nabil.zemiti]@lirmm.fr

camera undergoes movements while the surface undergoes deformation, "reconstruction and tracking" methods are thus preferred. Among such methods, VolumeDeform [9] stands out as the most proficient in recent years. It simultaneously tracks and reconstructs a deforming object with an handheld camera based on RGB and D information. However, it relies on color and geometric feature detection which are unfortunately not available when considiring applications with flat (without corner and edges), slightly textured surfaces typical in applications such as bioprinting or additive manufacturing.

In this work, we propose to perform precise 3D non-rigid motion estimation directly on a colored PC, without pre-constructed model. Our method is compatible with eye-in-hand ECL configuration as it does not require any mesh construction and is adapted to the changing and restricted field of view unlike state-of-the-art methods [5], [7], [8]. By employing 3D PCs, we can register them within the same 3D coordinate system given the camera movements.

Additionnally, we propose to use both the whole intensity of the scene as a dense representation and its depth making it suitable for flat slightly color-textured surfaces commonly encountered in bioprinting applications [3], as opposed to the findings presented in [9] which uses sparse features . The whole image intensity is exploited to encode displacements within the camera's observation plane, while the depth component is leveraged to recover 3D motion parameters. This combination enables the extraction of 3D non-rigid non-isometric transformation which would hardly be deduced from an RGB-only or a depth-only sensor.

The surface deformation is modeled as a 3D TPS function enabling its application to any PC, and guaranteeing regularization of the surface shape. A global overview of the method is depicted in Fig. 2. The steps of the proposed method allow the direct estimation of 3D non-rigid motion of flat slightly textured surfaces with an RGB-D camera in eye-in-hand configuration. To the best of authors knowledge, the overall method is novel in the context of eye-in-hand visual servoing for performing tasks on deforming objects.

The present paper is organized as follows. In Section II, the considered problem is explained, then the 3D non-rigid motion estimation method is presented in Section III and the overall process is finally validated experimentally on simulated dataset and a deforming silicon phantom in Section IV.

II. PROBLEM STATEMENT

The considered problem is depicted in Fig. 1. The goal is to follow a path on a surface which is deforming in a non-rigid non-isometric manner. It is assumed that the initial 3D path was defined on the surface at an instant t_0 with an RGB-D camera, in a reference frame C_0. It consists in a sequence of desired positions and orientations: ${}^{C_0}\mathcal{P}_d(t_0) = \{{}^{C_0}\mathbf{e}_{d,1}(t_0), {}^{C_0}\mathbf{e}_{d,2}(t_0), ..., {}^{C_0}\mathbf{e}_{d,N}(t_0)\}$ and ${}^{C_0}\mathcal{N}_d(t_0) = \{{}^{C_0}\mathbf{n}_{d,1}(t_0), {}^{C_0}\mathbf{n}_{d,2}(t_0), ..., {}^{C_0}\mathbf{n}_{d,N}(t_0)\}$ where N is a strictly positive integer that represents the number of reference points, and ${}^{C_0}\mathbf{e}_{d,k}(t_0), {}^{C_0}\mathbf{n}_{d,k}(t_0) \in \mathbb{R}^3$ are the desired tool positions and orientations respectively. This path constrains 5 degrees of freedom (DOF), which is enough for the considered applications as bioprinting. However, since the considered surface has a variable shape over time, the path and surface normals should be updated during the task.

To this end, the RGB-D camera in eye-in-hand ECL configuration is used. This setup enables the camera to simultaneously capture the end-effector and the target. This approach eliminates the errors related to the calibration of the robot-camera kinematic relationship. However, as the camera pose w.r.t. the robot is fixed, the camera is moving with the robot during the task and the view of the surface is partial and changing, making the use of a pre-constructed mesh impossible.

The camera movements can be obtained either thanks to the robot kinematics or an external tracker acquisition. Thus, the acquisitions which are expressed in a moving reference frame noted C_i can be expressed in C_0. Then, they have to be compared to the first acquisition in the corresponding region of interest in order to compute the surface deformation. This 3D deformation can be decomposed in an in-plane motion (due to expansion, retraction or rigid movement) and a vertical motion. Those two components are simultaneously contained in the RGB-D data and can thus be used to update the desired path.

An analytic expression of the deformation is necessary to be applicable to the initial desired path. As explained in [10], a 3D deformation can be defined as a TPS warp $w_{3D} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, which associates for each point **p** in a given PC, its coordinates **p'** in the deformed PC with respect to the coordinates of the TPS control points $\mathbf{c'}(t) = \begin{bmatrix}\mathbf{c}'_x(t) & \mathbf{c}'_y(t) & \mathbf{c}'_z(t)\end{bmatrix}^T$ with $\mathbf{c}'_x(t), \mathbf{c}'_y(t), \mathbf{c}'_z(t)$ respectively the vector of x, y, z coordinates of each control points at time t. That is to say $\forall \mathbf{p} \in \mathbb{R}^3, \mathbf{p'} = w_{3D}(\mathbf{p}, \mathbf{c'}(t))$. Thus, to transform the inital desired path to be adapted to the current deformed surface, **c'**(t) must be estimated. At each time, the first step is to estimate $\mathbf{c}'_x(t)$ and $\mathbf{c}'_y(t)$ based on intensity information, then $\mathbf{c}'_z(t)$ is deduced with the depth information.

III. NON-RIGID MOTION ESTIMATION

Since the camera displacement from the first to current pose ${}^{C_0}T_{C_i}$ is supposed to be available through an external tracking system or the robot kinematics, all the acquisitions are expressed in C_0. Considering that the current observed area is partial, the initial global acquisition should be cropped in the corresponding Region of Interest (ROI) by removing points with x-coordinates outside the minimum and maximum x-values of the ROI's vertices, and similarly for the y-coordinates.. This ROI is defined in the current frame C_i as a square of fixed coordinates under the end-effector and is placed in C_0 through ${}^{C_0}T_{C_i}$ (Fig. 2(b)).The two clouds, expressed in the C_0 coordinate system, are then cropped

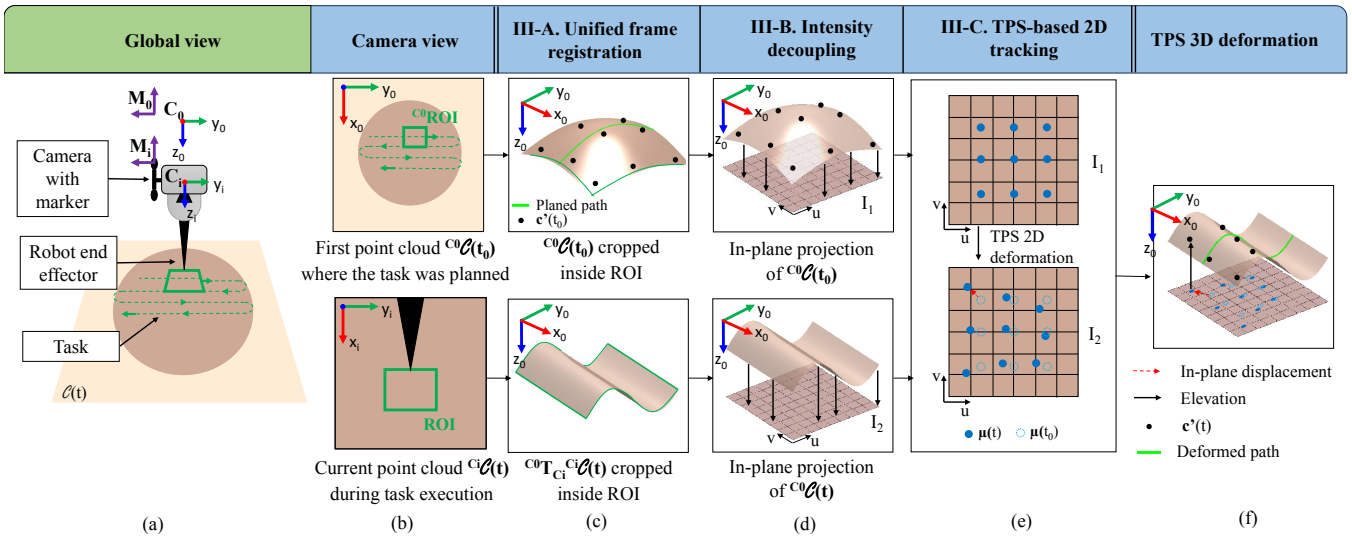

Fig. 2. Schematical overview of the proposed method: (a) The robot equipped with a camera and marker is following a path on a moving surface (b) At $t = t_0$ ${}^{C_0}\mathscr{C}(0)$ is acquired, and during task, the camera and surface moves, and ${}^{C_i}\mathscr{C}(t)$ is acquired (c) The region under the tool is registered in the initial reference frame (d) The intensity information is decoupled from the depth information (e) The in-plane deformation is computed through TPS-based tracking (f) The 3D deformation is deduced. Each step is explained in a dedicated section of the present paper.

inside that ROI and compared to get the surface deformation (Fig. 2(c)). To do so, their depth and intensity are decoupled in order to transform a PC, which is a discrete 3D surface data structure, into data easier to process. The intensity is used to recover in-plane deformation while depth is used to recover the surface elevation. Section III-A details how intensity and depth data are decoupled to be used to compute the surface deformation. Then, Section III-B explains the computation of the desired parameters.

A. Intensity decoupling

The displacement of each point of the studied surface due to its deformation is decoupled along two different directions. The in-plane displacement is computed based on camera acquisitions along its corresponding plane. In order to compute the planar displacement along $\boldsymbol{x_0}$ and $\boldsymbol{y_0}$ (which are the axes of C_0 as denoted in Fig. 2), the intensity information will be used. Thus, the PC, expressed in C_0 and noted $\mathscr{C}$ is flattened in $(\boldsymbol{x_0}, \boldsymbol{y_0})$ plane and each original colored 3D point $\mathbf{p}$ is converted to a grayscale 2D point $\mathbf{p}_{2D}$, i.e. $\forall \mathbf{p} = \{x, y, z, r, g, b\} \in \mathscr{C}, \mathbf{p}_{2D} = \{x, y, 0, I\}$ with $I = 0.299r + 0.587g + 0.114b$. The points in the PC are not regularly spaced, but it is crucial for effective processing and tracking to have uniformely spaced data. Thus, the resulting flattened PC is transformed into an intensity map of resolution r, where the intensity of each pixel is defined as the average of the intensities of the points in the bin (Fig. 3). That is to say, for each point $(x, y, 0, I)$ of the flattened PC, its coordinate in the generated image is $(u, v) = (\lfloor \frac{x - x_{min}}{r} \rfloor, \lfloor \frac{y - y_{min}}{r} \rfloor)$, with $x_{min}, y_{min}, x_{max}, y_{max}$ the limits of the ROI, and its value is the mean intensity of points having this coordinate. This process leads to a maximum error of $\frac{r}{2}$ mm in the estimation of x, y. An inherent issue with depth camera is the presence of holes due to reflective or homogeneous surface. To cope with this problem, during the intensity map computation, those holes are filled with color interpolated from the surrounding values. At each iteration, two images, noted I_1 and I_2, are created from cropped ${}^{C_0}\mathscr{C}(t_0)$ and ${}^{C_0}\mathscr{C}(t)$ within the ROI. With ${}^{C_0}\mathscr{C}(t_0)$ and ${}^{C_0}\mathscr{C}(t)$ the PC acquired by the camera at time t_0 and t and expressed in C_0 as depicted in Fig. 2(d).

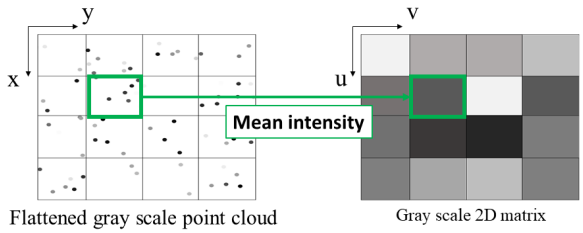

Fig. 3. Illustration of image formation principle: each bin is filled with the mean intensity values of 3D points inside.

B. 2D TPS warp estimation

To estimate the coordinates of the control points in millimeter in the camera's frame $\begin{bmatrix}\hat{\mathbf{c}_x}'(t) & \hat{\mathbf{c}_y}'(t)\end{bmatrix}^T$, the first step is to compute the coordinates of the control points in pixel in the current created image I_2 (see Section III-A) $\boldsymbol{\mu}(t) = \begin{bmatrix}\hat{\mathbf{c}_u}'(t) & \hat{\mathbf{c}_v}'(t)\end{bmatrix}^T$. To do so, we aim to find the 2D deformation $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ between I_1 and I_2 which is governed by a 2D TPS warp of control points $\boldsymbol{\mu}(t)$ as presented in [11] and illustrated in Fig. 2(e). Besides, a global bias λ and a multiplicative factor for each pixel are included in order to compensate for the change in the illumination between two image acquisitions as proposed in [12]. Each multiplicative factor is modeled as the interpolated value of a TPS function $w_{il} : \mathbb{R}^2 \rightarrow \mathbb{R}$ which associates for each

point in the image a corrective factor, based on the value of its control points [12]. Those points are placed in a regular 3×3 grid on the ROI and their associated values are noted $\boldsymbol{\mu_{il}}$. Therefore, defining a gray-scale 2D image as a function $I : \mathbb{R}^2 \rightarrow \mathbb{R}$,where $I(\mathbf{p})$ represents the intensity of pixel $\mathbf{p} = [u, v]^T$, the in-plane registration error is defined as

$$E(\boldsymbol{\mu}, \boldsymbol{\mu_{il}}, \lambda) = \sum_{\mathbf{p}=(u,v)} ||w_{il}(\mathbf{p}, \boldsymbol{\mu_{il}}) I_2(w(\mathbf{p}, \boldsymbol{\mu})) + \lambda - I_1(\mathbf{p})||^2 \quad (1)$$

The Gauss-Newton algorithm is used to minimize E with respect to $\boldsymbol{\mu}, \boldsymbol{\mu_{il}}$ and λ. The initial $\boldsymbol{\mu}$ is defined as a regular 3×3 grid on I_1 and $\boldsymbol{\mu_{il}}$ as the vector $[1]_{9\times1}$ and λ=0. Iterations are done until the number of iteration is superior to 200 or the variation of E is inferior to 0.001.

The real coordinates $\begin{bmatrix} \hat{\mathbf{c}_x}'(t) & \hat{\mathbf{c}_y}'(t) \end{bmatrix}^T$ of the control points are then obtained reversing the process presented in Section III-A. The 3D coordinates of the control points $\mathbf{c'}(t) = \begin{bmatrix} \mathbf{c}'_x(t) & \mathbf{c}'_y(t) & \mathbf{c}'_z(t) \end{bmatrix}^T$ are computed by finding the intersection between the 3D line which passes through each control point parallel to the vector $\boldsymbol{z_0}$ =[0 0 1], and the observed PC [13]. The inital desired path positions and normals $({}^{C_0}\mathcal{P}_d(t_0), {}^{C_0}\mathcal{N}_d(t_0))$ are thus deformed through the induced 3D TPS warp presented in Section II (Fig. 2(f)). At each ROI displacement, the previous result, shifted by the camera motion, is used as a warm start of the optimization algorithm.

IV. EXPERIMENTS AND RESULTS

The present section evaluates two main aspects of the proposed method. First, the obtained accuracy is assessed using simulated deformations. Since the surface displacements are known in this scenario, the non-rigid registration can be compared to a precise ground-truth. Additionally, the applicability is assessed using real deformations obtained with an actuated silicone phantom. The used robotic arm is a Panda (Franka Emika®). The external localizer FusionTrack 500 (Atracsys®) was used for accurate camera displacement estimation (0.9 mm precision up to 2m) as it has shown better performances in pose estimation than the Panda (Franka Emika®) (0.1 mm repetability). The RGB-D camera is an IntelRealSense D405 (Intel®). This camera uses a stereoscopic system to compute depth map and has the advantage to work at very short distance (7 cm). The proposed algorithm was implemented in C++ and was ran with an Intel®Core™i7-6820HQ CPU @ 2.70 GHz. We consider the context of *in situ* bioprinting [14] on burn injuries and breathing motion compensation for those experiments.

A. Simulated dataset

A challenging simulated dataset was used with a relatively small amount of color features, and a flat surface without corner or edges. The visual aspect of a burn injury was reproduced onto a white textile surface, and this setup was captured to create a PC representation $\mathscr{C}(t_0)$. The acquired PC was subjected to controlled deformations mimicking the characteristics of breathing motions using a $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ TPS function controlled by a regular grid of nine control points placed on $\mathscr{C}(t_0)$: the six points on left and right were sinusoidally moved along the **x**-axis with a 3 mm amplitude. The six points on the upper and lower part were sinusoidally moved along the **y**-axis with a 6 mm amplitude. And the center point was moved along the **z**-axis with a 10 mm amplitude. The chosen amplitudes exceeds typical breathing motions [15] to assess the performance of the algorithm in a quite challenging scenario. The period was set to 0.3 s and time step to 0.01 s which leads to a maximum in-frame displacement of 0.6, 1.3 and 2.1 mm along **x**, **y** and **z** direction respectively. The illumination factors in the sequence were adjusted using a $\mathbb{R}^2 \rightarrow \mathbb{R}$ TPS function controlled by nine control points which darkens the cloud when it expires and brightens it when it inspires to mimic real conditions. The values of control points were determined to maintain color values within [0,255].

Besides, a moving ROI of size 27x27 pixels with resolution 0.7 mm based on the RGBD camera depth map resolution, representing the restricted field of view of the camera, was defined on the initial PC and moves from the left to the right, covering the burn injury area. Within this moving ROI, a set of 9 control points was uniformly arranged in a grid pattern. Each successive pair of points is spaced 9 mm apart. These control points were subject to the same deformation as the PC, thereby establishing the ground truth control points placement $\mathbf{c}'_{GT}(t)$.A subset of the generated clouds and ROI is depicted in Fig. 4.

The method described in Section III is then applied on the sequence of 250 PCs and camera positions, and the retrieved $\mathbf{c'}(t)$ are compared to $\mathbf{c}'_{GT}(t)$.

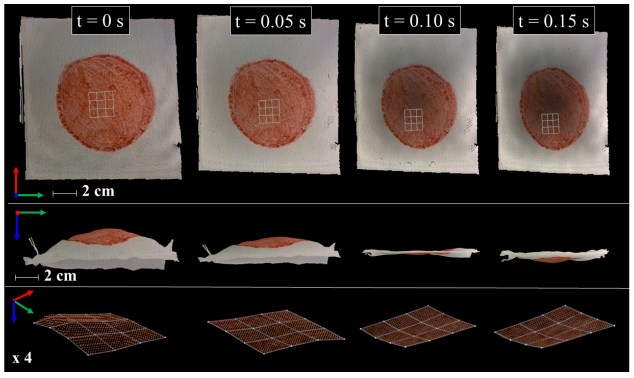

Fig. 4. Subset of simulated PCs and ROI result: the first row contains an upper view of the deformed PC where the in-plane deformation and ROI displacement can be seen. The **z**-axis deformation can be seen in the second row. The third row shows the obtained 3D deformed ROI.

The mean global distance between found and ground truth control points is 0.07 mm with a maximum of 0.3 mm (Fig. 5) which means that the x, y, z deformations were well estimated. The residual error E (Eq. 1) is in mean 1.8 on 256 gray-level (i.e. 0.7%).

For example, the resulting deformation of control point n°5, placed in the center of the ROI as illustrated in Fig.

5, is depicted in Fig. 6(a) and is compared with the ground truth motion in Fig. 6(b). Camera motion was subtracted from the global point motion so the deformation-only pattern, especially the amplitude and frequency of the imposed deformation, can be recognized. It has to be noted that due to the dynamic movement of the ROI across the surface and the distinct deformation of each surface point, the resulting measurements in the **x**, **y**, and **z** dimensions do not represent a pure sine function. As a matter of fact, a precise sine function would be obtained only for the original control points defined in C_0.

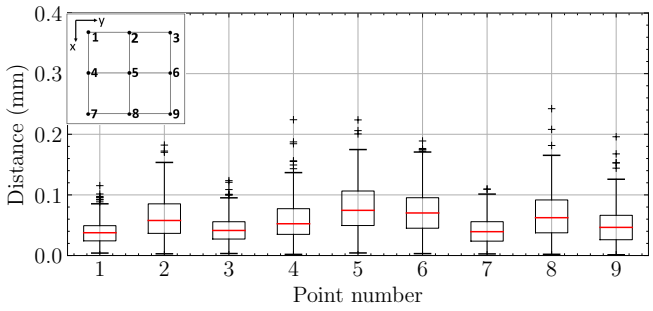

Fig. 5. Box whiskler plot representing the distance from ground truth to computed data for the nine control points in the ROI.

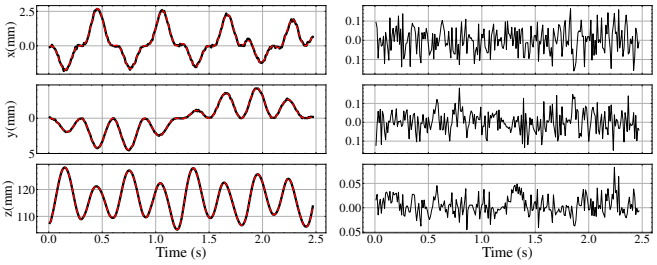

Fig. 6. Evolution of the 5th control point coordinate in simulated dataset: (a) represents the control point displacement and (b) is the distance from ground thruth.

B. Real dataset

An actuated silicone phantom was fabricated to induce non-rigid motions. This platform behaves as a balloon, undergoing non-rigid deformations through controlled inflation using a 200 ml syringe operated by a stepper motor, which simulates the action of a breathing machine. This platform is fixed next to a robotic arm holding a camera in eye-in-hand configuration, so that the endpoint of the end-effector is always visible. A trajectory is planned offline so that the robotic arm, equipped with the camera, moves above the silicon balloon mimicking a bio-printing task while the silicon platform is deformed (Fig. 7). An external optical localizer is placed alongside to track the motion of the camera. This setup allows to validate the motion estimation algorithm in realistic conditions.

The obtained mean residual error E on 256 gray-levels is 2.8 (i.e. 1.09%). Note that this value has a similar order

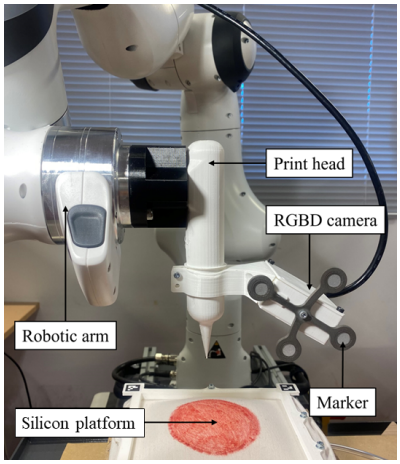

Fig. 7. System overview: the robotic arm holds a 3D-printed model of a print head and a camera equipped with a marker.

of magnitude to the residual error obtained in Section IV.A. This indicates that the computed registration is reliable. In spite of the presence of noise and regions not populated with colored points (an inherent issue in completely textureless surface with RGB-D camera using stereoscopic perception to compute depth), the results are consistent as the ROI was defined to be in the center of the camera field of view (Fig.8). The resulting control point n°5 motion is illustrated in Fig.9 and was separated from the camera motion as explained before. The results are similar to the simulated ones, and are in accordance with the design of the silicon platform actuation.

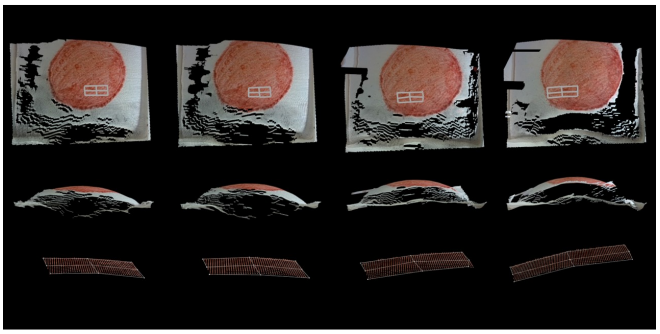

Fig. 8. Subset of real clouds and ROI result: the first row contains and upper view of the deformed cloud where the in-plane deformation and ROI displacement can be seen, the z-axis deformation can be seen in the second row, the third row show the obtained 3D deformed ROI. The complete sequence can be seen in the attached video

In the context of the current study, computation time for each frame was observed to be approximately 0.7 s on a 50x50 = 2500 pixels image with 9 control points. While this processing time appears relatively high for robot control applications, it is important to note that this is mostly due to the optimization part described in Section III-B. Indeed, the image formation described in Section III-A takes on average

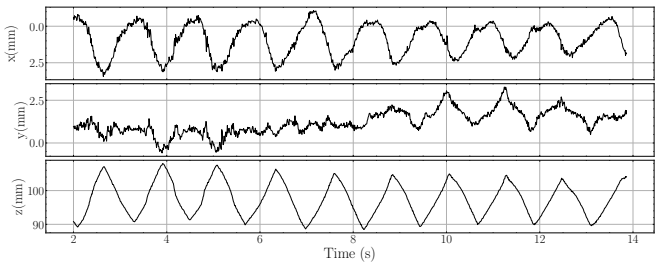

Fig. 9. Deformation of 5th control point coordinate in real dataset

0.25 ms. However, the computation of the jacobian and the transformed image, which necessitates the multiplication of matrices $n \times n$ with n the number of pixel in the considered image (in our study, 2500 pixels) is done at each Gauss-Newton iteration, implying an increased computation time that needs to be reduced. Notably, the use of a Graphics Processing Unit (GPU) could improve the computation time with parallelized matrix calculation, and the use of a more efficient optimization algorithm represents a short-term goal for refining the method's real-time capabilities. Nevertheless, it is important to highlight that the proposed method does not involve a computationally intensive step such as mesh generation, which can last 1 min [8].

Another limitation is that the method faces challenges recovering the in-plane displacement from intensity for completely uniform surfaces, as is typical for all vision-based methods when dealing with wholly homogeneous surface (which are still rare in surgical context). Besides, a stereoscopic RGB-D camera was used for data acquisition and this kind of technology struggles to reconstruct non-textured surface as it can be seen in Fig.9 in the white part of the surface. Thus, considering the incorporation of a short-range structured-light camera, which are more precise, could offer a beneficial solution. Even those cameras are sensitive to specular reflection, if they are sparse they can easily be treated with hole filling by interpolation, or by removing it from the Gauss-Newton minimization as shown in [12]. Additionnaly, the incorporation of prediction methods to deal with occlusions and anticipate motion could enhance the performances.

V. CONCLUSION

The proposed non-rigid motion estimation method allows for retrieving a surface non-rigid motion with submillimeter accuracy with RGB-D data. It was designed to be adapted to the context of eye-in-hand robotic visual servoing and dealing with non-rigid motion of flat slighlty textured surfaces while the camera is moving. The method relies on the entire intensity of the image and depth data to recover 3D deformations, rather than utilizing color or geometric features. This approach proves beneficial for medical applications like bioprinting, where surfaces typically exhibits flat and uniform characteristics. Ultimately, the methodology presented herein is intended to be integrated into a robotic control framework, facilitating the compensation of non-rigid motions in eye-in-hand configuration. In this regard, the attained error is considered satisfactory. Future works should then enhance the computation time, robustify the estimation through predictive methods and integrate it into a robotic control framework.

REFERENCES

[1] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Li, Jia Pan, Wenzhen Yuan, and Michael Gienger. Challenges and Outlook in Robotic Manipulation of Deformable Objects. *IEEE Robotics & Automation Magazine*, 29(3):67–77, September 2022.

[2] Ezgi Kucukdeger and Blake N. Johnson. Closed-loop controlled conformal 3d printing on moving objects via tool-localized object position sensing. *Journal of Manufacturing Processes*, 89:39–49, 2023.

[3] Zhijie Zhu, Shuang-Zhuang Guo, Tessa Hirdler, Cindy Eide, Xiaoxiao Fan, Jakub Tolar, and Michael C. McAlpine. 3D Printed Functional and Biological Materials on Moving Freeform Surfaces. *Adv. Mater.*, 30(23):1707495, June 2018.

[4] Andreas Jordt and Reinhard Koch. Direct Model-Based Tracking of 3D Object Deformations in Depth and Color Video. *Int J Comput Vis*, 102(1):239–255, March 2013.

[5] Antoine Petit, Vincenzo Lippiello, and Bruno Siciliano. Real-time tracking of 3D elastic objects with an RGB-D sensor. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3914–3921. IEEE, September 2015. Place: Hamburg, Germany.

[6] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.

[7] Agniva Sengupta, Alexandre Krupa, and Eric Marchand. Visual Tracking of Deforming Objects Using Physics-based Models. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14178–14184. IEEE, May 2021. Place: Xi'an, China.

[8] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics*, 33(4):1–12, July 2014.

[9] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 362–379, Cham, 2016. Springer International Publishing.

[10] Deukhee Lee and Alexandre Krupa. Intensity-based visual servoing for non-rigid motion compensation of soft tissue structures due to physiological motion using 4d ultrasound. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2831–2836, 2011.

[11] Jongwoo Lim and Ming-Hsuan Yang. A Direct Method for Modeling Non-Rigid Motion with Thin Plate Spline. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1196–1202. IEEE, 2005. Place: San Diego, CA, USA.

[12] Rogério Richa, Philippe Poignet, and Chao Liu. Three-dimensional Motion Tracking for Beating Heart Surgery Using a Thin-plate Spline Deformable Model. *The International Journal of Robotics Research*, 29(2):218–230, February 2010.

[13] Phillip N. Azariadis and Nickolas S. Sapidis. Drawing curves onto a cloud of points for point-based modelling. *Computer-Aided Design*, 37(1):109–122, January 2005.

[14] João Cavalcanti Santos, Lenaïc Cuau, Philippe Poignet, and Nabil Zemiti. Decoupled model predictive control for path following on complex surfaces. *IEEE Robotics and Automation Letters*, 8(4):2046–2053, 2023.

[15] T Wentz, H Fayad, J Bert, O Pradier, J F Clement, S Vourch, N Boussion, and D Visvikis. Accuracy of dynamic patient surface monitoring using a time-of-flight camera and B-spline modeling for respiratory motion characterization. *Phys. Med. Biol.*, 57(13):4175–4193, July 2012.