Localization and Tracking of User-Defined Points on Deformable Objects for Robotic Manipulation

Sven Dittus, Benjamin Alt, Andreas Hermann, Darko Katic and Rainer Jäkel
ArtiMinds Robotics, Karlsruhe, Germany
{sven.dittus | benjamin.alt | andreas.hermann | darko.katic | rainer.jaekel}@artiminds.com

Jürgen Fleischer
Institute of Production Science, Karlsruhe Institute of Technology, Germany
juergen.fleischer@kit.edu

***Abstract*—This paper introduces an efficient procedure to localize user-defined points on the surface of deformable objects and track their positions in 3D space over time. To cope with a deformable object's infinite number of DOF, we propose a discretized deformation field, which is estimated during runtime using a multi-step non-linear solver pipeline. The resulting high-dimensional energy minimization problem describes the deviation between an offline-defined reference model and a pre-processed camera image. An additional regularization term allows for assumptions about the object's hidden areas and increases the solver's numerical stability. Our approach is capable of solving the localization problem online in a data-parallel manner, making it ideally suitable for the perception of non-rigid objects in industrial manufacturing processes.**

I. Introduction

Many manufacturing processes rely on image processing to enable industrial robots to manipulate objects. Whereas many sophisticated camera systems meet the need for localizing and tracking user-defined Points of Interest (POIs) on rigid objects, there is still no sufficiently accurate solution for coping with this problem for deformable objects yet. Moreover, existing approaches reconstruct the deformable object‘s model online, requiring POIs to be defined at runtime and thus being unsuitable for fully automated processes. This paper proposes a solution for defining POIs on an offline model and then localizing and tracking these points on a deformable object in an online detection pipeline.

II. Related Work

Existing approaches for localizing and tracking POIs are only applicable to specific object categories (e.g. linear [1] or planar [2], [3]), assume speficic deformation models (e.g. articulated models [4] or skeletons [5]) or particular materials (e.g. textiles [6], [7]) or are restricted to detecting specific features (e.g. points on corners or edges [8], [9]). To reach the precision required for sophisticated manipulation tasks, prior work requires external markers [10], [11] or elaborate physics models [3], [12]–[15]. We combine several SotA-solutions for rigid object state estimation (such as SHOT descriptors), physical modelling (such as deformation fields) and computer graphics (such as projection) into a processing pipeline which permits the tracking and localization of user-defined points on arbitrary deformable objects using only a single depth camera without markers or prior physics modeling.

This work was partially supported by the German Federal Ministry of Education and Research (BMBF) under the grant no. 16SV8406.

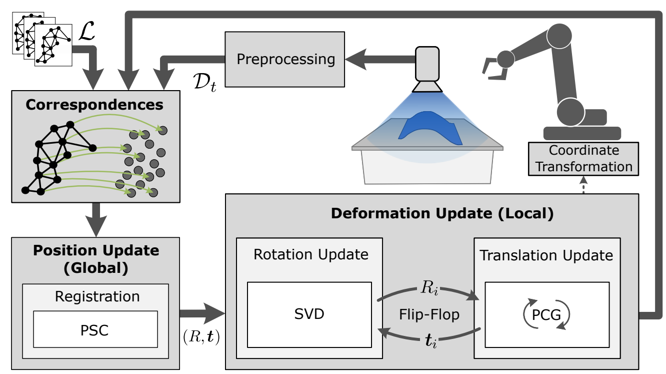

Fig. 1: Overview of the localization and tracking pipeline

III. Method

Our algorithm consists of three distinct phases: (1) **Demonstration** of the reference model and the POIs; (2) an **iterative localization and tracking process** consisting of **observing** a new point cloud, identification of **correspondences** between the observation and the deformed reference model of the previous timestep and **estimation of the deformation**; and (3) a **coordinate transformation** of the localized POIs into the robot end-effector coordinate system for subsequent manipulation.

A. Surface and deformation model

To efficiently perform computations, we model object surfaces as triangle meshes, while deformations are modelled via a *deformation grid* [16]. Unlike [16], we use a highly detailed mesh as a surface representation which is independent of the deformation model's resolution. This allows to increase computation performance while maintaining a highly detailed surface. Our deformation model consists of two data structures, both containing $|G|$ grid points. Whereas the equally spaced

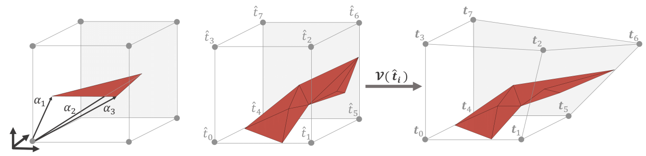

Fig. 2: Trilinear weights (l.); static and deformed grid cells (r.)

static grid G describes the undeformed reference model, the deformation field $\mathcal{V}$ represents the object's deformed state at the current timestep t. Each gridpoint i is defined in G by a position vector $\hat{\boldsymbol{t}}_i$, allowing to express the position $\hat{\boldsymbol{p}}$ of an undeformed vertex within G as $\hat{\boldsymbol{p}} = \sum_{i=1}^{|G|} \alpha_i \hat{\boldsymbol{t}}_i$ with trilinear weights $\alpha_i \in [0, 1]$ (cf. fig. 2). The position $\boldsymbol{p}$ of the same vertex in the deformation field $\mathcal{V}$ can be described analogously by a weighted sum of deformed gridpoint positions $\boldsymbol{t}_i$ as $\boldsymbol{p} = \sum_{i=1}^{|G|} \alpha_i \boldsymbol{t}_i$. We define an *observation* $\mathcal{D}$ as an organized point cloud of the deformed object.

B. Demonstration

Most SotA approaches use either a low-resolution reference model of the deformable object [17] or none at all [16], [18]. To allow for user demonstrations of POIs, our algorithm requires a high-detail reference model to be created offline. To improve the stability of our solver, we limit the object's initial deformation with respect to its reference model by demonstrating a library $\mathcal{L}$ of reference models in an offline step. Each model in $\mathcal{L}$ is a triangle mesh of the object in a distinct deformation state. The models are generated by fusing several depth images into a Truncated Signed Distance Field (TSDF) and then extracting the triangles via the MarchingCubes algorithm [19]. After the demonstration of the reference models, the user can select relevant POIs on the meshed surface via a graphical user interface. At runtime, after the first observation $\mathcal{D}$, the model in $\mathcal{L}$ most similar to $\mathcal{D}$ is used to initialize G.

C. Correspondence identification

The definition and identification of *correspondences* links the current observation and the deformed reference model of the previous timestep $(t-1)$ and forms the basis of the deformation estimation: The estimation of a deformation is equivalent to the minimization of the distances between all correspondences. We define three correspondence types:

a) Point-to-point (P2P): Result from projecting each surface point $\boldsymbol{p}_c$ of the deformed reference model into the image plane and comparing it to the corresponding point $\boldsymbol{p}_c^a$ that has been measured. The quality of a P2P-correspondence can be described by a weight $w_c = (\frac{w_d + w_n + w_v}{3})^2$ where w_d denotes the distance between projected point $\boldsymbol{p}_c$ and its correspondent $\boldsymbol{p}_c^a$, w_n the distance between $\boldsymbol{p}_c$'s normal $\boldsymbol{n}_c$ and its correspondent $\boldsymbol{n}_c^a$, and $\boldsymbol{w}_v$ the angle between the camera view direction $\boldsymbol{v}$ and $\boldsymbol{n}_c$.

b) Point-to-surface (P2S): Projective correspondences such as P2P typically only yield approximate, not exact, correspondences. P2S-correspondences add another degree of freedom by associating a point in the model to a *plane* in the observation, defined by the P2P-correspondence $\boldsymbol{p}_c^a$ and its normal $\boldsymbol{n}_c^a$. During deformation estimation, this allows to only minimize the distance d_c along the normal.

c) Feature correspondences: Unlike projective correspondences, correspondences based on feature matching can detect large deformations, tangential movements and rotations of the object out of the image plane. Prior work [20], [21] and our own experiments have found the PFH and FPFH descriptors to be highly sensitive and specific but to scale poorly with the size of the point cloud, while SHOT descriptors scale linearly and are robust against outliers. We implement feature correspondences using SHOT, as its sensitivity suffices for most real-world applications.

D. Deformation estimation

The deformation of the reference object can be estimated by formulating an optimization problem to estimate their degrees of freedom and thus the deformation of the reference object. Using the notation introduced in III-A, the deformation of a single grid point i can be expressed as $\mathcal{V}_i = \hat{\boldsymbol{t}}_i - \boldsymbol{t}_i$. For estimating the deformation field, we split up all unknows into a single global rigid transformation $(\boldsymbol{t}, R)$ and many local transformations $(\boldsymbol{t}_i, R_i)$ and combine them in a vector $\mathcal{X}$:

$$\mathcal{X} = \left(\boldsymbol{t},\ R,\ \underbrace{\ldots, \boldsymbol{t}_i^T, \cdots}_{3\ |G|\ translations} \mid \underbrace{\cdots, R_i, \cdots}_{3\ |G|\ rotations} \right)^T \tag{1}$$

The interpretation of correspondences as error terms E allows to formulate the deformation estimation of $\mathcal{X}$ as an energy minimization problem, which is also suggested by [16]–[18]. This optimization can be regarded as a model regression problem and solved by existing solvers:

$$\begin{aligned} E(\mathcal{X}) = &\ \omega_p E_{P2P}(\mathcal{X}) + \omega_s E_{P2S}(\mathcal{X}) \\ &+ \omega_f E_F(\mathcal{X}) + \omega_r E_{Reg}(\mathcal{X}) \end{aligned} \tag{2}$$

[17] and [16] solve a similar high-dimensional nonlinear optimization problem by linearizing the model and using the Gauss-Newton method, incurring a significant overhead for the computation of the Jacobian J. [16] splits the optimization into a two-stage process composed of a *fixed registration* followed by a *deformation estimation*. We leverage the fact observed in [22] that the deformation estimation can again be split into two independent sub-problems, which allows to solve for nonlinear rotations and linear translations using iterative Gauss-Newton on each subproblem in turn ("flip-flop" strategy). We perform fixed registration, the estimation of a global transformation $(\boldsymbol{t}, R)$, via Prerejective RANSAC (PSC). For the deformation estimation, setting up the Jacobian for the error terms of the three correspondence types is straightforward:

$$E_{P2P} = \sum_{c=1}^{|C|} \omega_c \left\| \underbrace{\underbrace{R \left[\sum_{i=1}^{|G|} \alpha_i(\hat{\boldsymbol{p}}_c)\ \boldsymbol{t}_i \right] + \boldsymbol{t}}_{\mathcal{V}(\hat{\boldsymbol{p}}_c)} - \boldsymbol{p}_c^a}_{\boldsymbol{r}_{P2P,C}(\mathcal{X})} \right\|_2^2 \tag{3}$$

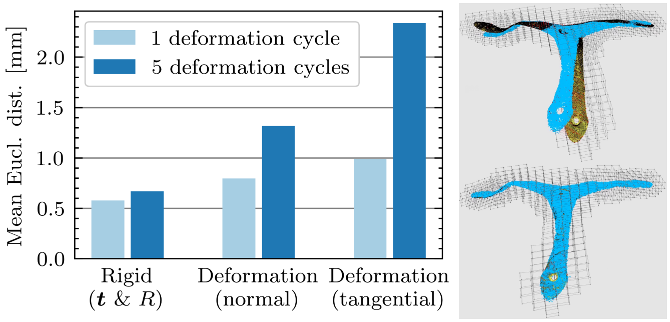

Fig. 3: Precision for different deformation types (l.); estimated deformation grid and ground-truth point cloud (blue) before and after one deformation cycle (r.)

$$J_{P2P,ci} = \frac{\partial\left(\omega_p\ \omega_c\ \boldsymbol{r}_{P2P,c}(\mathcal{X})\right)}{\partial \boldsymbol{t}_i} = \omega_p\ \omega_c\ \alpha_i(\hat{\boldsymbol{p}}_c) \quad (4)$$

where $|C|$ is the number of correspondences and $J_{P2P,ci}$ is the entry at the c^{th} row and i^{th} column of the Jacobian J_{P2P}. The Jacobians for E_{P2S} and E_F can be found analogously.

a) Regularization: With a single camera's perspective, it is impossible to observe the complete surface of an object. The spatial lack of correspondences implies an underdetermined equation system and E being ill-conditioned. To alleviate this problem, we use an ARAP regularizer [22], where non-observable surface points are deformed such that the total deformation of the body is *as rigid as possible*. Unlike prior work [22], [23], we estimate the deformation in terms of $\mathcal{V}$ instead of the mesh, leading to the adapted ARAP term

$$E_{Reg} = \sum_{i=1}^{|G|} \sum_{j \in \mathcal{N}_i} \left|\left| (\boldsymbol{t}_i - \boldsymbol{t}_j) - R_i\left(\hat{\boldsymbol{t}}_i - \hat{\boldsymbol{t}}_j\right) \right|\right|_2^2, \quad (5)$$

where $\mathcal{N}_i$ denotes the *neighborhood* (6 surrounding grid points) of grid point $i \in [1, |G|]$. For each grid point i, our solver must solve for 6 unknowns describing its pose $(R_i|\boldsymbol{t}_i)$. As shown in [23], the (non-linear) estimation of R_i can be solved in closed form given $\boldsymbol{t}_i$. For $\boldsymbol{t}_i$, we obtain

$$\frac{\partial E_{Reg,i}}{\partial \boldsymbol{t}_i} \stackrel{!}{=} 0 \Leftrightarrow \underbrace{\sum_{j \in \mathcal{N}_i} (\boldsymbol{t}_i - \boldsymbol{t}_j)}_{L\ \cdot\ \mathcal{X}_i} = \sum_{j \in \mathcal{N}_i} \frac{R_i + R_j}{2} (\hat{\boldsymbol{t}}_i - \hat{\boldsymbol{t}}_j) \quad (6)$$

where the left-hand side is the product of the Laplace matrix L with the vector of all unknowns $\mathcal{X}_i$.

b) Flip-flop solver: We iteratively estimate R_i and $\boldsymbol{t}_i$ in turn by closed-form solving for R_i via singular value decomposition (see [23] for details) and approximating $\boldsymbol{t}_i$ via Gauss-Newton, where the update step $\Delta\mathcal{X}$ is obtained via preconditioned conjugate gradients (PCG). Using the Jacobians derived above, we can obtain the deformation $\mathcal{X}_{t+1}$ after an update step via

$$J^T J := J_{P2P}^T J_{P2P} + J_{P2S}^T J_{P2S} + J_F^T J_F + L^T L \quad (7)$$

$$J^T \boldsymbol{r} := J_{P2P}^T \boldsymbol{r}_{P2P} + J_{P2S}^T \boldsymbol{r}_{P2S} + J_F^T \boldsymbol{r}_F + L^T \boldsymbol{r}_{Reg} \quad (8)$$

$$J^T J\ \ \boldsymbol{\Delta}\mathcal{X} = J^T \boldsymbol{r} \quad (9)$$

$$\mathcal{X}_{t+1} = \mathcal{X}_t + \boldsymbol{\Delta}\mathcal{X} \quad (10)$$

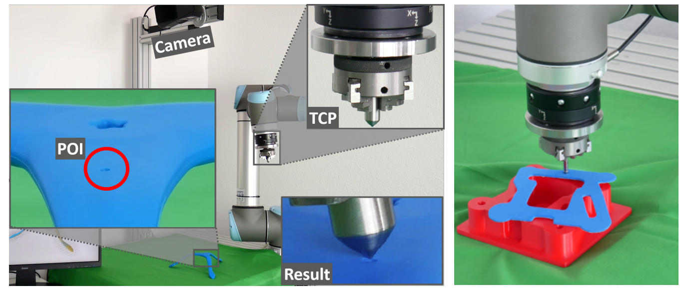

Fig. 4: POI tracking on a deformable tripod (l.); rubber seal assembly (r.)

IV. Results

a) Precision: In a first set of experiments, we assess the precision of our approach by comparing tracking results versus manually labeled correspondences. 10 POIs on a deformable tripod were considered, with each POI also fitted with a color-coded marker to facilitate manual labeling.[1] The tripod was repeatedly deformed and the poses of the POIs were estimated by our algorithm as well as via the markers (cf. fig. 3). Our approach was capable of *localizing* all POIs with sub-millimeter accuracy, and *tracking* all POIs with errors between 0.6 and 2.2 mm. Unlike feature-matching based approaches, we always estimate the deformation of the complete surface and thereby avoid "mismatching" POIs by design.

b) Performance: A significant advantage of our approach is that each step of the solver pipeline can be efficiently parallelized. We benchmarked our algorithm using reference and deformation models at fine[2] and coarse[3] resolutions. A parallelized CPU implementation of our algorithm localized all POIs in under $1.5s$ in both cases on consumer hardware.

c) Robot experiments: In a first robot experiment, we track a point on the surface of a tripod subjected to several deformations of up to 20% of the tripod's arm length, or ca. 2.5 cm. We use a UR5 robot equipped with a measuring tip to visualize tracking results (cf. fig. 4 (l.)), confirming precision within 2 mm. In a second experiment, we use our approach to position a flat rubber seal on a housing, illustrating its potential for real-world industrial applications (cf. fig. 4 (r.)).

V. Discussion and Outlook

Our approach and solver pipeline allows efficient tracking and localization of POIs on deformable objects. Where prior work requires markers, explicit modelling or does not allow for offline POI definition, our approach achieves sub-millimeter precision localization and millimeter-precision tracking without these drawbacks. This makes it particularly suitable for applications in industrial robotics and flexible, quickly reconfigurable assembly or surface treatment tasks. We are working on integrating our solution into an industrial robot manipulation framework, a more efficient GPU implementation and a more extensive evaluation on a wider set of benchmarks.

[1] Since our algorithm only considers geometric features, the presence of the markers neither helped nor hurt the algorithm.

[2] Reference model: 30000 vertices, $\mathcal{V}$: 3250 grid points

[3] Reference model: 15000 vertices, $\mathcal{V}$: 700 grid points

REFERENCES

[1] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3426–3433, 2018.

[2] M. Tang, R. Tong, R. Narain, C. Meng, and D. Manocha, "A GPU-based streaming algorithm for high-resolution cloth simulation," *Comput. Graph. Forum*, vol. 32, no. 7, pp. 21–30, 2013.

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

[4] T. Schmidt, R. Newcombe, and D. Fox, "DART: Dense articulated real-time tracking with consumer depth cameras," *Auton. Robots*, vol. 39, no. 3, pp. 239–258, 2015.

[5] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1746–1753.

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

[7] Y. Li, Y. Wang, Y. Yue, D. Xu, M. Case, S.-F. Chang, E. Grinspun, and P. K. Allen, "Model-driven feedforward prediction for manipulation of deformable objects," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1621–1638, 2018.

[8] K. Yamazaki, "Grasping point selection on an item of crumpled clothing based on relational shape description," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3123–3128.

[9] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1703–1708.

[10] J. Finnegan and J. Dorsey, Eds., *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*. New York, New York, USA: ACM Press, 2006.

[11] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, "Total capture: 3D human pose estimation fusing video and inertial sensors," in *Procedings of the British Machine Vision Conference 2017*, T.-K. Kim, S. Zafeiriou, G. Brostow, and K. Mikolajczyk, Eds. British Machine Vision Association, 2017.

[12] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: Speeded up robust features," in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 3951, pp. 404–417.

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

[14] H. Lang, J. Linn, and M. Arnold, "Multi-body dynamics simulation of geometrically exact Cosserat rods," *Multibody Syst. Dyn.*, vol. 25, no. 3, pp. 285–312, 2011.

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

[16] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, "VolumeDeform: Real-time volumetric non-rigid reconstruction," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, vol. 9912, pp. 362–379.

[17] M. Zollhöfer, C. Theobalt, M. Stamminger, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, and C. Loop, "Real-time non-rigid reconstruction using an RGB-D camera," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, 2014.

[18] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," *IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, pp. 343–352, 2015.

[19] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.

[20] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, 2016.

[21] R. Hänsch, T. Weber, and O. Hellwich, "Comparison of 3D interest point detectors and descriptors for point cloud fusion," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. II-3, pp. 57–64, 2014.

[22] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, ser. SGP '07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 109–116.

[23] ——, "Least-squares rigid motion using SVD," Zürich, 2017.