Tracking an elastic object with an RGB-D sensor for a pizza chef robot

Antoine Petit, Vincenzo Lippiello, Bruno Siciliano[1]

***Abstract*— This paper presents a method to track in real-time a 3D object which undergoes large deformations such as elastic ones, and fast rigid motions, using the point cloud data provided by a RGB-D sensor. This solution would contribute to robotic humanoid manipulation purposes. Our framework relies on a prior visual segmentation of the object in the image. The segmented point cloud is then registered first in a rigid manner and then by non-rigidly fitting the mesh, based on the Finite Element Method to model elasticity and on geometrical point-to-point correspondences to compute external forces exerted on the mesh. The real-time performance of the system is demonstrated on real data involving challenging deformations and motions, for a pizza dough to be ideally manipulated by a chef robot.**

I. Introduction

Whereas tracking problems for rigid objects using vision sensors has reached a certain maturity, perception for non-rigid objects is a challenging problem. It has aroused much interest in recent years in the computer vision, computer graphics and robotics communities. A lot of potential applications would indeed be targeted, in fields such as augmented reality, medical imaging, robotic manipulation, by handling a huge variety of objects like tissues, paper, rubber, viscous fluids, cables, food, organs, etc.

This study comes within the scope of the RoDyMan project[2], consisting in a unified framework for robotic dynamic manipulation of deformable objects. As seen in Figure 1, a demonstration scenario is the humanoid dual-arm/hand manipulation of the pizza dough, in an authentic manner, showing a humanoid robot involved in culinary traditions and rituals.

With respect to rigid objects, the problem of dealing with deformations poses several additional challenges such as modeling the properties of the considered material, and fitting this model with the vision and/or range data. This registration problem also involves critical real-time concerns, which are especially required for robotic dynamic manipulation. Although numerous studies have proposed efficient real-time techniques to handle 3D surfaces (paper, clothes) which undergo isometric or slightly elastic deformations, a large open field remains when considering larger elastic deformations. The aim of this paper is thus to propose a real-time tracking system able to handle elastic deformable objects, as well as fast rigid motions, using visual and range data, through an RGB-D sensor.

Fig. 1: Overview of the RoDyMan robotic platform and the pizza making process.

II. Related works and motivations

In the literature, the various approaches proposed to register deformable objects, using vision and/or range data, could be classified according to the underlying model of the considered object and its physical realism.

A. Registration using implicit physical modeling

Based on implicit physical models, approaches in [10, 1, 13] use a 1D parametric curve or 2D splines models (B-splines, Radial Basis Functions) to track deformable objects in monocular images. This class of methods relies on the minimization of an energy function involving an external energy term related to some image features, and an internal energy term regularizing curvature, bending or twisting, compelling the model to vary smoothly. Adapting these techniques to register with 3D shapes or surfaces in monocular images is much more complex, since 3D deformations can imply ambiguous 2D transformations, resulting in an underconstrained problem. A first attempt by Terzopoulos *et al.* [17], relying on 3D splines and inspired by [10], densely processes gradient features, to compute the data energy term. Feature-based approaches [15], less ambiguous, have been preferred and additional constraints are often added to solve ambiguities. With point cloud data, methods in [9, 18] employ a RGB-D sensor to register the surface mesh to the acquired point cloud by minimizing, an error function accounting for geometric or direct depth and color errors, and a stretching penalty function for the mesh. By means of a NURBS parametrization [9] or an optimized GPU implementation [18], real-time performances can be achieved.

B. Registration using explicit physical modeling

Instead, another formulation of the problem relies on explicit physics-based deformable models to perform registration, by modeling more explicitly elasticity. With respect to implicit methods, other sorts (such as non-linear elasticity) and magnitudes of deformations can be treated, inferring more consistently shape and/or volumetric regularization. Statically, the solution can then be determined, by setting

[1] A. Petit, V. Lippiello and B. Siciliano are with DIETI, Università degli Studi di Napoli Federico II, Italy, {antoine.petit, vincenzo.lippiello, bruno.siciliano}@unina.it

[2] http://www.rodyman.eu/

internal and external forces equal or, equivalently, minimizing energy functions. Physics-based methods include discrete mass-spring-damper systems [4, 16], or more explicit approaches relying on the Finite Element Method (FEM), based on continuum mechanics. In [16], based on mass-spring-damper systems, 3D-3D correspondences, determined through a probabilistic inference, enable the computation of the external forces applied to the mesh. First attempts for registration employing the FEM for 3D surfaces in [3, 12] used linear elasticity FEM models. More recently, in [11], registration in monocular images is addressed by designing a stretching/shrinking energy using continuous mechanical constraints on 2D elements assuming linear elasticity, and some 3D boundary conditions. Haouchine *et al.* [7] uses a linear tetrahedral co-rotational FEM model, coping with larger elastic deformations, external forces being related to correspondences between tracked 3D feature points mapped to the 3D mesh, by means of a stereo camera system.

C. *Motivations and contributions*

Since our system would attempt to handle large deformations and elastic volumetric strains, a realistic mechanical model, based on the FEM, has been adopted. Besides, for potential robotic dynamic manipulation applications, an explicit physical modeling would enable the reliable computation and prediction of internal forces undergone by the object and thus to perform proper force control tasks. The recent suitability of these models for real-time applications, as demonstrated in [16, 7], has confirmed our choice. Robustness concerns, for instance with regard to textureless objects, have lead us to rely on an RGB-D sensor. Among the methods having the closest goals, motivations and constraints to ours, we could mention [9, 16, 7, 18]. With respect to them, several contributions are proposed, such as handling various large deformations such as elastic ones, handling fast rigid motions, handling occlusions, and addressing all these tasks in real-time (40 fps).

III. Segmentation

In this work we advocate the use of a prior segmentation step in order to restrict the acquired point cloud to the object of interest, see section V for a more detailed justification.

A. *Grabcut segmentation*

We rely here on the efficient and widespread *GrabCut* segmentation technique [14], based on graph cuts. In its original formulation, the *Grabcut* algorithm addresses the visual bilayer segmentation task as an energy minimization problem, based on statistical models of the foreground (the object) and the background.

For an image, we denote by $\alpha = \{\alpha_i\}_{i=1}^N$ the set of the unknown binary labels of the set of pixels ($\alpha_i = 0$ for the background pixels, $\alpha_i = 1$ for the foreground). Estimating the values $\widehat{\alpha}$ of the labels can be formulated as the minimization of an energy-based Markov Random Field objective function $E(\alpha)$, with respect to α:

$$E(\alpha) = E_{data}(\alpha) + \gamma E_{smooth}(\alpha) \quad (1)$$

$$\text{with} \quad E_{data}(\alpha) = \sum_i U_i(\alpha_i) \quad (2)$$

E_{data} is the data energy term, with $U_i(\alpha_i)$ accounting for the observation probability for a pixel to belong to the foreground or to the background, based on some image "data" (intensity, color, location...) observed on the pixel, using the statistical models built for the background and the foreground. E_{smooth} is the smoothness energy term whose goal is to favor smoothness, or spatial coherence within the pixels.

In order to compute the optimal solution of this energy minimization problem and to determine $\widehat{\alpha}$, a *graph cuts* minimization algorithm [2] is employed.

Statistical models for the data energy function are Gaussian Mixture Models (GMM) based on color distributions, learned for both the foreground and background layers, which are initially determined by the user through a bounding box around the foreground on the initial image. Besides, pixels outside this bounding box are definitely assigned to the background layer ($U_i(\alpha_i = 0) = inf$), whereas inside their label is unknown, so that energy minimization only has effects inside the bounding box.

B. *Temporal coherence and real-time issues*

Once the initial image is segmented through user interaction, the following frames are treated by updating the area to effectively segment. As shown in Figure 2, the silhouette contour of the previous segmented foreground is extracted, and the distance transform is computed over it, providing a signed distance map to these contours. According to a fixed threshold on this distance map, we define a narrow strip around the contour, in which labels of the pixels are unknown (grey area on 2(d)), whereas they are definitely assigned to the foreground on the inner side of the strip ($U_i(\alpha_i = 1) = inf$, white area on 2(d)), and to the background otherwise ($U_i(\alpha_i = 0) = inf$). In this manner, temporal consistency is ensured, since energy minimization is only effective within this strip, in the vicinity of the previous segmentation boundary, avoiding some outliers outside or inside, and reducing significantly computations.

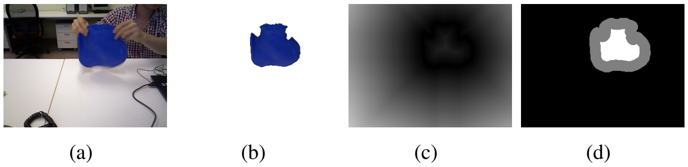
(a) (b) (c) (d)

Fig. 2: Temporal consistency for segmentation. Segmentation is effective on the strip (grey area on 2(d)) around the contour of the previous segmented frame (2(b)), through the distance map to the contour (2(c)).

IV. Deformable object modeling with the Finite Element Method

Since we deal with objects which may undergo large elastic deformations, a major issue lies in the definition of a relevant physical model. The Finite Element Method

(FEM) provides a realistic physical model, by relying on continuum mechanics, instead of finite differences for mass-spring systems for instance. It consists in tessellating the deformable object into a mesh made of elements, usually tetrahedrons. The deformation field $\mathbf{u}_e$ over an element e is then approximated as a continuous interpolation of the displacements $\hat{\mathbf{u}}_e$ of its vertices. We rely here on a volumetric linear FEM approach with tetrahedral elements.

By resorting to the infinitesimal strain theory and linear elasticity through Hooke's law, the internal elastic forces $\mathbf{f}_e$ exerted on the four vertices of e of the mesh can be linearly related to their displacements:

$$\mathbf{f}_e = \mathbf{K}_e \hat{\mathbf{u}}_e \tag{3}$$

with $\mathbf{K}_e$ the 12×12 stiffness matrix of the element e.

Although it is insensitive to translation transformations, the model, by using an infinitesimal approximation of the strain tensor, is however inaccurate when modeling large rotations of the elements, leading for instance to non-zero summations of the forces. A work-around consists in the co-rotational approach [5], used for registration purposes in [7], and which is a good compromise between the ability to model large linear elastic deformations and computational efficiency. Since the displacement of an element can be decomposed into a rigid transformation and a pure deformation, the idea is to extract the rotation matrix $\mathbf{R}_e$ related to the rigid transformation. Then the stiffness matrix can be warped with respect to this rotation, so as to accommodate to rotation transformations, giving:

$$\mathbf{f}_e = \mathbf{R}_e \mathbf{K}_e (\mathbf{R}_e^{-1} \mathbf{x}_e - \mathbf{x}_{e,0}) \tag{4}$$

with $\hat{\mathbf{u}}_e = \mathbf{x}_e - \mathbf{x}_{e,0}$, $\mathbf{x}_e$ and $\mathbf{x}_{e,0}$ being respectively the current and initial positions of the vertices of e.

In this way, the overall forces on the whole mesh can be summed to zero, while computational efficiency is ensured since $\mathbf{K}_e$ can be computed in advance, in contrast to non-linear FEM approaches.

V. Registration with point cloud data

Our deformable registration problem consists in fitting the point cloud data provided by an RGB-D sensor, with the tetrahedral mesh. The basic idea is to derive external forces exerted by the point cloud on the mesh and to integrate these forces, along with the internal forces computed using the physical model presented in section IV, into a numerical solver solving the resulting mechanical equations.

In this work, these external forces are computed based on geometrical point-to-point correspondences between the point cloud and the mesh, relaxing the assumption of having a textured object [7]. We assume that the mesh is available (manually designed here) and correctly initialized. Let us however note that off-line automatic reconstruction and meshing techniques could be considered to build the mesh and initialization could be addressed through some learning and recognition of spin images [8] or local 3D features. Besides, the Young modulus and Poisson ratio of the considered material are assumed to be known.

A. Segmented point cloud

As introduced in section III, we use the acquired RGB image sequence to visually segment the object of interest from its background and occlusions. Since we do not rely on some distinctive visual features, the point could provided by the depth sensor has indeed to be restricted to the considered object, to avoid ambiguities with the background or with occluding shapes. This point cloud is thus segmented using the visual segmentation.

B. Rigid iterative closest point

A first step in our method is to register the segmented point cloud in terms of rigid translation and rotation transformations, initially considering the mesh of the object as rigid. We thus suggest a classical rigid Iterative Closest Point (ICP) algorithm between the segmented point cloud and the vertices of the visible surface of the mesh. With this procedure, which converges rapidly, fast rigid motions can be tracked and a fair initialization for the non-rigid process can be obtained.

C. Deformable iterative closest point

In order to register the segmented point cloud with the mesh in a non-rigid manner, we suggest an ICP-like procedure. By means of Kd-trees searches, nearest neighbor correspondences are determined, both from the segmented point cloud to the visible surface of the mesh and from the visible surface of the mesh to the segmented point cloud.

From the segmented point cloud to the mesh, correspondences are more suited to track expansion deformations under stretching forces, and they are insensitive to occlusions and segmentation errors. Conversely, from the visible surface of the mesh to the segmented point cloud, correspondences are instead more suited to track shrinking deformations under compression, while being however sensitive to occlusions.

Then the points lying on the visible surface of the mesh, stretching elastic forces $\mathbf{f}_{ext}$ are computed, from the mesh-to-point cloud correspondences, and, to a lesser extent, from the point cloud-to-mesh correspondences, since our application would mainly deal with stretching efforts, and since occlusions are an issue we intend to deal with. Estimating the deformations of the mesh consists in solving a dynamic system of non-linear ordinary differential equations involving the internal and external forces, based on Lagrangian dynamics:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{f} = \mathbf{f}_{ext} \tag{5}$$

where $\mathbf{M}$ is the mass matrix, and $\mathbf{C}$ the damping matrix, and $\mathbf{f}$ assembling the element-wise forces $\mathbf{f}_e$.

An Euler implicit integration scheme is used to solve the system, along with a conjugate gradient method. In case of severe deformations, correspondences initially established may not be very consistent, therefore the procedure is iteratively repeated, up to a fixed number of iterations.

VI. Experimental results

In order to carry out experiments, the point cloud data is acquired from a calibrated RGB-D camera Asus Xtion, 320 × 240 RGB and depth images being processed. A standard laptop with an NVIDIA GeForce 720M graphic card has been used, along with a 2.4GHz Intel Core i7 CPU. Since fast real-time performances are required, the segmentation process relies on a CUDA implementation. For the non-rigid registration phase, we have employed the Simulation Open Framework Architecture (SOFA) [6], which enables to deal with various physical model to evolve simulations in real-time. The results presented here deal with a pizza-like elastic object. It undergoes fast rigid motions and deformations such as large elastic ones, similar to the ones involved in the pizza making process, in the scope of the RoDyMan project. Qualitative results are presented in Figure 3. On the first row are shown input RGB images, the second features the corresponding segmented frame and finally the third row shows the 3D mesh tracking the object. We can notice (see also attached video) the ability of the process to correctly segment the visible part of the object, to track fast rigid motion and to accurately register deformations, while being robust to occlusions due to the hands manipulating the object. The whole algorithm runs on the sequence at around 40 *fps*, with a mesh made of 280 points and point clouds with similar resolutions.

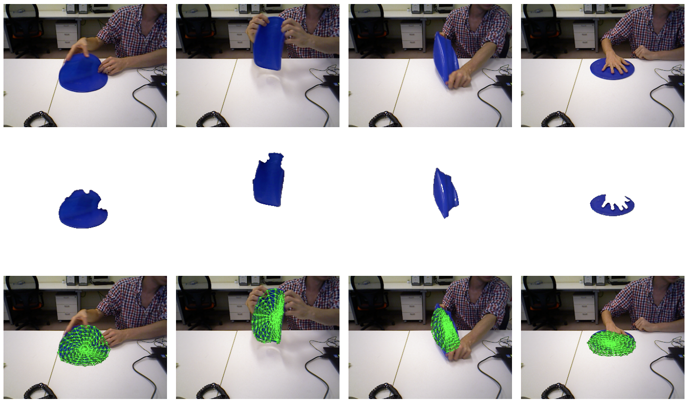

Fig. 3: Results of the tracking process, with the input images (first row), the segmented frames (second), and the registered mesh reprojected in the input image.

VII. Conclusion

The recent development of physics-based modeling methods for deformable elastic objects for registration purposes and the availability of real-time implementations have lead us to choosing such an approach to track an object subjected to various large deformations, with a RGB-D sensor. The use of a pertinent linear FEM model, of an efficient segmentation method, and of classical point cloud registration techniques have made our system a promising real-time tracking method able to handle various deformations and motions. Future works would aim at improving the point cloud matching procedure and extend the whole process to other deformations such as plastic ones, as well as demonstrating the suitability of the approach to mime the art of making pizzas with a dual arm/hand robot.

References

[1] Adrien Bartoli, Andrew Zisserman, et al. Direct estimation of non-rigid registrations. In *British Machine Vision Conference*, pages 899–908, 2004.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 1222–1239, November 2001.

[3] Laurent D Cohen and Isaac Cohen. Deformable models for 3-d medical images using finite elements and balloons. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 592–598. IEEE, 1992.

[4] Christof Elbrechter, Robert Haschke, and Helge Ritter. Bi-manual robotic paper manipulation based on real-time marker tracking and physical modelling. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1427–1432. IEEE, 2011.

[5] Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. A fast finite element solution for cloth modelling. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, pages 244–251. IEEE, 2003.

[6] François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, et al. Sofa: A multi-model framework for interactive physical simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pages 283–321. Springer, 2012.

[7] Nazim Haouchine, Jeremie Dequidt, Igor Peterlik, Erwan Kerrien, Marie-Odile Berger, and Stephane Cotin. Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pages 199–208. IEEE, 2013.

[8] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[9] Andreas Jordt and Reinhard Koch. Direct model-based tracking of 3d object deformations in depth and color video. *International Journal of Computer Vision*, pages 1–17, 2013.

[10] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[11] Abed Malti, Richard Hartley, Adrien Bartoli, and Jae-Hak Kim. Monocular template-based 3d reconstruction of extensible surfaces with local linear elasticity. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1522–1529. IEEE, 2013.

[12] Tim McInerney and Demetri Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 518–523. IEEE, 1993.

[13] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *Int. Journal of Computer Vision*, 76(2):109–122, February 2007.

[14] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314, 2004.

[15] Mathieu Salzmann, Julien Pilet, Slobodan Ilic, and Pascal Fua. Surface deformation models for nonrigid 3d shape recovery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8), 2007.

[16] John Schulman, Alex Lee, Jonathan Ho, and Pieter Abbeel. Tracking deformable objects with point clouds. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1130–1137. IEEE, 2013.

[17] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial intelligence*, 36(1):91–123, 1988.

[18] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics, TOG*, 2014.