HAL
archives-ouvertes.fr

Real-time tracking of 3D elastic objects with an RGB-D sensor

Antoine Petit, Vincenzo Lippiello, Bruno Siciliano

▶ **To cite this version:**

Antoine Petit, Vincenzo Lippiello, Bruno Siciliano. Real-time tracking of 3D elastic objects with an RGB-D sensor. IROS 2015 - IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Sep 2015, Hamburg, Germany. pp.3914-3921, 10.1109/IROS.2015.7353928 . hal-01617309

HAL Id: hal-01617309
https://hal.archives-ouvertes.fr/hal-01617309

Submitted on 16 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Real-time tracking of 3D elastic objects with an RGB-D sensor

Antoine Petit, Vincenzo Lippiello, Bruno Siciliano[1]

***Abstract*— This paper presents a method to track in real-time a 3D textureless object which undergoes large deformations such as elastic ones, and rigid motions, using the point cloud data provided by an RGB-D sensor. This solution is expected to be useful for enhanced manipulation of humanoid robotic systems. Our framework relies on a prior visual segmentation of the object in the image. The segmented point cloud is 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 synthetic and real data involving challenging deformations and motions.**

I. Introduction

Unlike vision-based tracking problems with rigid objects, for which a certain maturity has been reached, perception for non-rigid objects is still 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: 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 Fig. 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 objects, potentially textureless, by tracking large deformations and fast rigid motions, using visual and range data provided by an RGB-D sensor.

To cope with such deformations, our approach involves a physical modeling of the considered object, by relying on a Finite Element Method (FEM). The considerable progresses recently made within the computer graphics and medical simulation domains have enabled real-time performance for processing such models. As demonstrated in this paper, our whole system is able to run fastly at around 35 frame per seconds.

The remainder of the paper is organized as follows. Some works related to ours are presented in Sect. II. Our system requires the visual segmentation of the object, what is addressed in Sect. III. Then in Sect. IV the mechanical model of the object considered here is introduced, Sect. V explains how the point cloud data is processed and matched with the model to perform registration. Finally, some experimental results are presented in Sect. VI.

Fig. 1: Artistic views 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. Let us first clarify our scope and distinguish it from non-rigid reconstruction methods for which at each frame provided by the vision/range sensor, a single mesh is reconstructed, as in [1, 23, 17]. Instead, the goal is here to continuously estimate the rigid transformations and the deformations undergone by a specific object, modeled by a known mesh.

A. Registration using implicit physical modeling

Based on implicit physical models, approaches in [13, 2, 18] 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

[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/ The research leading to these results has been supported by the RoDyMan project, which has received funding from the European Research Council (FP7 IDEAS) under Advanced Grant agreement number 320992. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

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.* [22], relying on 3D splines and inspired by [13], densely processes gradient features, to compute the data energy term. Less ambiguous feature-based approaches such as [20] have been preferred and additional constraints are often added to solve ambiguities. With point cloud data, methods in [12, 24] employ an RGB-D sensor to register the acquired point cloud to a surface mesh 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 [12] or an optimized GPU implementation [24], real-time performance can be achieved. Although these two systems have shown promising and impressive results, they are still bounded to isometric or slightly elastic deformations, by means of regularization functions proportional to squared distances between nodes of the mesh, whereas we wish to model elastic in more physically realistic manner, to handle larger strains. Another limitation of these methods is that they process mesh to input point cloud correspondences in their data error functions, and are thus sensitive to missing data, or unobserved areas of the considered object due to occlusions. We consider in this paper also correspondences from the input point cloud to the mesh, through the use of a segmentation method to restrict the input point cloud to the observed areas of the object, and based on these correspondences, the occluded or unobserved areas would not affect registration.

B. Registration using explicit physical modeling

Instead, another formulation of the problem relies on 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. Statistically, the solution can be determined, by setting internal and external forces equal or, equivalently, minimizing energy functions. Physics-based methods include discrete a mass-spring-damper system [14, 6, 21], or more explicit approaches relying on the Finite Element Method (FEM), based on continuum mechanics. In [21], 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 [4, 16] used linear elasticity FEM models. More recently, in [15], 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.* [10] 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. To the best of our knowledge, this latter method proposes the most realistic physical elastic model within a real-time vision-based tracking system, and we propose a similar model in this paper.

C. 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 by promising approaches [6, 21, 10], has confirmed our choice. We assume the prior knowledge of a consistent mesh (which could be automatically reconstructed offline) and of the material properties (through the Young modulus and the Poisson ratio), which could be estimated offline. Robustness concerns, regarding for instance 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 can mention [12, 21, 10, 24]. With respect to them, several contributions are proposed, such as handling various large deformations like elastic ones, handling rigid motions, handling occlusions, and addressing all these tasks in real-time (35 fps).

D. Overview of the system

As also represented in Fig.2, our tracking system can be outlined as follows: Input : the known 3D volumetric mesh of the object, a given RGB-D data , and assuming a fair registration at the previous time step.

1) Visual segmentation of the considered object, with a graph cut-based approach ensuring temporal coherence.
2) Using the resulted segmented point cloud, perform a rigid Iterative Closest Point (ICP) to estimate a rigid transformation from the point cloud to the mesh.
3) Using the resulting segmented point cloud, compute external linear elastic forces exerted on the vertices of the mesh from the point cloud to the mesh and conversely, based on closest point correspondences.
4) Compute elastic internal forces, based on a tetrahedral linear co-rotational FEM model.
5) Numerical resolution of mechanical equations.

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).

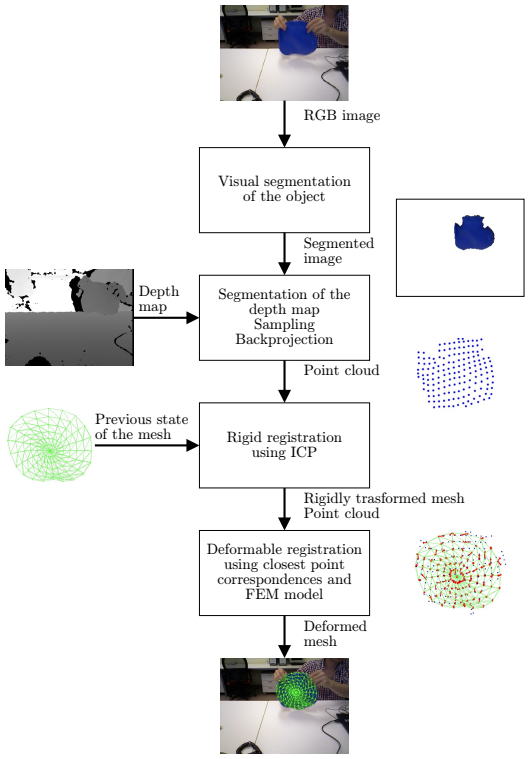

Fig. 2: Overview of our approach for deformable object tracking.

A. Grabcut segmentation

We rely here on the efficient and widespread *GrabCut* method [19], 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 input image $\mathbf{I}$, 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.

In order to compute the optimal solution of this energy minimization problem and determine $\widehat{\alpha}$, a *graph cuts* minimization algorithm [3] is employed, providing us with a segmented frame $\mathbf{I}^s$.

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 Fig. 3, 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 3d), whereas they are definitely assigned to the foreground on the inner side of the strip ($U_i(\alpha_i = 1) = inf$, white area on 3d), 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) Original frame (b) Segmentation (c) Distance map (d) Trimap

Fig. 3: Temporal consistency for segmentation. Segmentation will be effective on the strip (grey area on (d)) around the contour of the previous segmented frame (b), through the distance map to the contour (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) [5] 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:

$$\mathbf{f}_e = \mathbf{K}_e \hat{\mathbf{u}}_e \quad (3)$$

with being $\mathbf{K}_e$ the 12×12 stiffness matrix of the element e, depending on two elastic parameters of the material, the

Young modulus E, which measures the ratio between the tensile stress and the extensional strain and the Poisson ratio ν, which measures, under compression efforts on the object, the amount of expansion it undergoes in the two perpendicular directions [5].

Although it is insensitive to translation transformations, the model, by using an infinitesimal approximation of the strain tensor, giving a constant $\mathbf{K}_e$ linearizing the elastic forces, is however inaccurate when modeling large rotations of the elements, the non-linear effects leading to non-zero summations of the forces and causing for instance unexpected growth of volume. A work-around consists in the co-rotational approach [7], used for registration purposes in [10], which is a good compromise between the ability to model large elastic deformations, cope 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 rotation transformations, giving:

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

with being $\hat{\mathbf{u}}_e^r = \mathbf{R}_e^{-1}\mathbf{x}_e - \mathbf{x}_{e,0}$, with $\mathbf{R}_e^{-1}\mathbf{x}_e$ the back rotated deformed coordinates of the vertices of e, to an unrotated frame, the forces $\mathbf{K}_e\hat{\mathbf{u}}_e^r$ being then rotated to the current deformed element through the multiplication by $\mathbf{R}_e$. 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 Sect. 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 [10] with a rough surface, for which 2D/3D keypoints can be extracted and matched. 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 [11] or local 3D features. Besides, the Young modulus and Poisson ratio of the considered material are assumed to be known.

A. Segmented and sampled point cloud

As introduced in Sect. 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 cloud provided by the depth sensor is indeed restricted to the considered object so as to avoid ambiguities in the matching process with the background or with occluding shapes, and to be able to process correspondences from the input point cloud to the mesh. Using the segmented image $\mathbf{I}^s$, a segmented depth map $\mathbf{D}^s$ is obtained by aligning and intersecting the original input depth map $\mathbf{D}$ with the segmented area in $\mathbf{I}^s$. Then by back-projecting $\mathbf{D}^s$ in the sensor frame, the desired segmented point cloud $Y = \{\mathbf{y}_j\}_{j=1}^{n_Y}$, with $\mathbf{y}_j$ a 3D point in the sensor frame, is determined. For computational concerns, we limit the size of Y by first sampling $\mathbf{D}^s$ on a regular grid in the image plane.

B. Rigid iterative closest point

A first step in our method is to register the observed segmented point cloud Y in terms of rigid translation and rotation transformations, initially considering the mesh of the object as rigid. Let us first define $X = \{\mathbf{x}_j\}_{j=1}^{n_X}$ the set of vertices of the mesh, in its previous computed state. We suggest a classical rigid ICP algorithm between Y and the vertices of the visible surface X_V of the mesh transformed with respect to the previous RGB-D data. X_V is determined by performing a visibility test on the rendered depth map of the projected 3D mesh of the object. Through 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.

1) Nearest neighbor correspondences: By means of K-d tree 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. This step provides us with the sets of nearest neighbors $N_{X_V} = \{\mathbf{NN}_Y(\mathbf{x}_j) \mid \mathbf{x}_j \in X_V\}$ and $N_Y = \{\mathbf{NN}_X(\mathbf{y}_j)\}_{j=1}^{n_Y}$ respectively in Y for X_V, with the 1-NN function $\mathbf{NN}_Y$, and in X_V for Y, with the 1-NN function $\mathbf{NN}_{X_V}$.

Both sets of correspondences are processed since relying on the sole geometrical proximity may lead to inconsistent matches using single point-to-point matches.

Indeed from the segmented point cloud to the mesh, correspondences enable to track for instance expansion deformations under stretching forces, for which the observed segmented point cloud Y would spread over the visible surface of the mesh X_V. The extended areas of Y with respect to X_V can be matched with the outer areas of X_V. These correspondences also enable to deal with occlusions and segmentation errors since the corresponding unobserved areas of the object would not affect the underlying areas of X_V. Conversely, from X_V to Y, correspondences are instead more suited to track shrinking deformations under compression actions, the outer areas of X_V being coherently

matched with the outer areas of the observed point cloud Y of the compressed object. As a drawback, unobserved areas (occlusions, segmentation errors) would affect the underlying areas X_V which would match with the closest areas of Y.

As described hereafter, a trade-off has to be found between these two sets of correspondences, whether the application deals with stretching or compression actions on the object, and whether occlusions or segmentation errors are to be dealt with.

2) Computation of external forces: Based on the two sets of mesh-to-point cloud and point cloud-to-mesh correspondences, given by N_{X_V} and N_Y, an external elastic force $\mathbf{f}_{ext}$ exerted on each $\mathbf{x}_j$ in X_V, can be computed as follows:

$$\mathbf{f}_{ext}(\mathbf{x}_j) = k(\mathbf{x}_j - \mathbf{y}_j^f) \tag{5}$$

with

$$\mathbf{y}_j^f = \begin{cases} \lambda \mathbf{NN}_Y(\mathbf{x}_j) + (1-\lambda)\frac{1}{n_{K_j}} \sum_{\mathbf{y}_j \in K_j} \mathbf{y}_j \\ \quad \text{if } \sharp K_j > 0 \\ \\ \lambda \mathbf{NN}_Y(\mathbf{x}_j) + (1-\lambda)\mathbf{x}_j \\ \quad \text{if } \sharp K_j = 0 \end{cases} \tag{6}$$

where $K_j = \{\mathbf{y}_i \in Y | \mathbf{NN}_{X_V}(\mathbf{y}_i) = \mathbf{x}_j\}$ is the set of points in Y whose nearest neighbors are $\mathbf{x}_j$; k is the stiffness of these external elastic forces. As in [9], it is set to the same order of magnitude of the Young modulus, to be physically consistent. The fixed scalar λ tunes the balance between the mesh-to-point cloud and point cloud-to-mesh correspondences, as a trade-off suggested above (section V-C.1) between the stretching or compression actions to be tracked. If K_j is empty, the missing point cloud-to-mesh correspondences are replaced by a self-contribution for the vertice $\mathbf{x}_j$, compelling it to remain at its current position. In Fig.4, the vectors $\mathbf{x}_j - \mathbf{y}_j^f$ are displayed, from each $\mathbf{x}_j$. Some outliers in the point cloud may result in aberrant correspondences and thus aberrant forces exerted on some vertices. A simple solution has been to discard points in the point cloud whose point-to-point distances with their nearest neighbors in the mesh are above a certain threshold with respect to the mean value and the standard deviation of the whole set of point-to-point distances. In this case, for the considered vertices $\mathbf{x}_j$, we have $\mathbf{f}_{ext}(\mathbf{x}_j) = 0$.

Finally, regarding points $\mathbf{x}_j$ in X which are not visible, we also set $\mathbf{f}_{ext}(\mathbf{x}_j) = 0$.

The whole set of forces is finally concatenated in a vector $\mathbf{f}_{ext}$ of size n_X.

3) Weighting forces using contours: A limitation of this method lies in tracking large elastic deformations due to stretching efforts for instance. In this case, since correspondences are established based on 3D geometry, only vertices lying on the outer contour of the mesh are attracted to the extended area in the point cloud. As a consequence, forces attracting the contours are weak. We propose to emphasize them by weighting the vertices of the visible surface of the mesh, given their distance to the occluding contour of the mesh in the image plane. Based on the depth map $\mathbf{d}^M$ of the projected mesh, we compute the distance map of the occluding contour of the mesh, determined by rendering the projection of the mesh in the image plane. Then, the weight w_j for the vertex $\mathbf{x}_j$ is computed as follows:

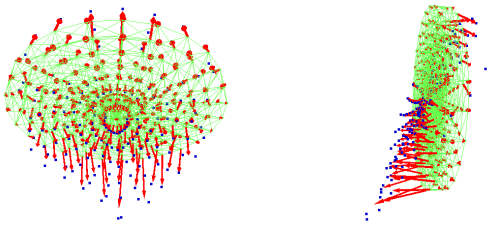

Fig. 4: External forces exerted on the vertices of the mesh, with $k = 1$.

$$w_j \propto e^{-\frac{d_j^M}{\sigma}} \tag{7}$$

where d_j^M is the distance from $\mathbf{x}_j$ to the nearest contour of the projected mesh, σ is a parameter which is empirically set; w_j is normalized so that we get an observation probability. Finally, forces are computed this way:

$$\mathbf{f}_{ext}(\mathbf{x}_j) = w_j k(\mathbf{x}_j - \mathbf{y}_j^f) \tag{8}$$

4) Numerical solver to compute the deformations: 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:

$$\mathbf{M}\ddot{\hat{\mathbf{u}}} + \mathbf{C}\dot{\hat{\mathbf{u}}} + \mathbf{f} = \mathbf{f}_{ext} \tag{9}$$

where $\mathbf{M}$ is the mass matrix, and $\mathbf{C}$ is the damping matrix, and $\mathbf{f}$ contains the element-wise forces $\mathbf{f}_e$. An Euler implicit integration scheme is used to solve the system, along with a conjugate gradient method. Using the estimated displacements $\hat{\mathbf{u}}$ of the vertices of the mesh, X can be updated. In case of severe deformations, correspondences initially established may not be very consistent, and thus the procedure is iteratively repeated, up to a fixed number K of iterations ($K = 3$ in the experiments presented in section VI).

VI. Experimental results

In order to evaluate the performance of our method and contributions, some experimental results are shown in this section, in a quantitative manner on some computer-generated data, and in a qualitative manner on some real data. Different objects, deformations and conditions are tested.

For the non-rigid registration phase, we have employed the Simulation Open Framework Architecture (SOFA) simulator [8], which enables to deal with various physical models and to evolve simulations in real-time.

A. Results on synthetic data

Relying on the SOFA framework, we have generated a sequence involving the deformations of a cylindrical elastic object, modeled by the FEM co-rotational approach. It has a Young Modulus of $E = 0.130MPa$ and Poisson ratio of $\nu = 0.49$. The tetrahedral mesh is made of 352 vertices with a circumferential/radial/height resolution of $25 \times 7 \times 2$, for radius/height dimensions of $0.11 \times 0.02m$, and is featured in Fig. 4. An impulse elastic stretching force is applied in the $-Z$ direction (see Fig. 5), on one point on the border of the object (point 1), for which few other points are fixed on the opposite border (points 4,5,6), and two compression forces are applied along Y and $-Y$ (points 2,3). The applied forces result in a fast elongation deformation of the object, with a maximum elongation above 50% at frame 25. For the tracking phase, segmentation aspects are not considered in these experiments. We only process the visible vertices of the rendered object in the sequence, and as a ground truth, the positions of the whole set of points are stored for evaluation. The following models and methods have been compared:

- Mass-spring model
- Standard FEM model, based on 3
- Co-rotational FEM model
- Co-rotational FEM model along with contour weighting (CW) (proposed method)

where for the mass-spring model, the vertices consist of point masses connected together by springs, deformations being solved through Newton's second law. Results can be visually observed in Fig. 5, and in Fig. 6 the 3D errors between the vertices of the registered mesh and the corresponding points in the point cloud are plotted (see also the attached video). The benefit of our method can also be observed in Fig. 5 featuring the original target (red) and the tracking 3D mesh (blue). The suitability of the co-rotational model can be stressed out and with the contour weighting technique, our method manages better to track the extensions on the extremities, even though some errors remain.

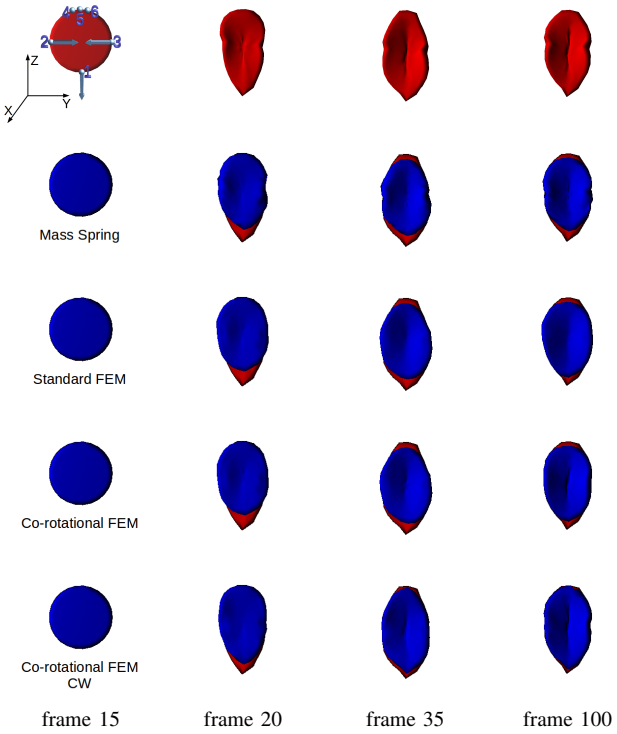

Fig. 5: Results of the deformable tracking process, for the cylindrical object. On the first row is featured the ground truth, on the second the tracking with the linear FEM, the third with the linear FEM and contour weighting, the fourth with the co-rotational FEM and the fifth adding contour weighting.

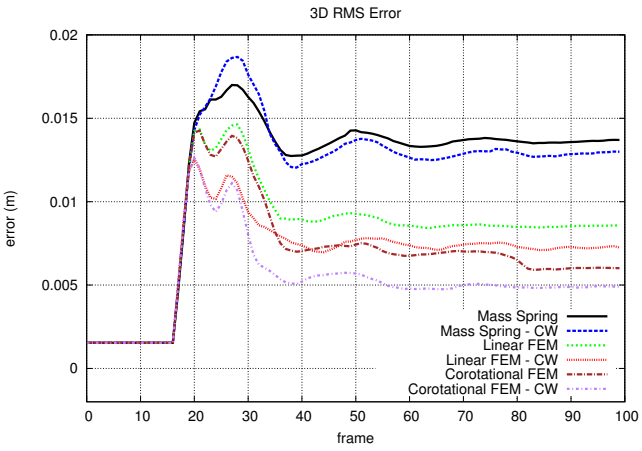

Fig. 6: Errors of the deformable tracking process, for the cylindrical object, with the different tested approaches. The results with contour weighting method for the mass spring and standard linear FEM models are also shown.

B. Results on real data

In order to carry out experiments on real data, the point cloud of the investigated scene 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. Here the segmentation process is involved in the loop, and since fast real-time performance is required, it relies on a CUDA implementation. The results presented here deal with a pizza-like elastic object, lacking of texture and showing a smooth surface, and with an elastic cylindrical bar object made with modeling clay.

For the pizza-like object, the idea has been to test motions and deformations similar to the ones involved in the pizza making process, in the scope of the RoDyMan project. On the presented sequence, the object undergoes fast rigid motions and various deformations such as isometric or elastic ones. The involved mesh has a circumferential/radial/height resolution of $25 \times 7 \times 2$, consisting in 352 vertices, as depicted in Fig. 8. The Young Modulus has been empirically set to $E = 0.150MPa$ and the Poisson ratio to $\nu = 0.3$. Qualitative results are presented in Fig. 7, comparing our method using the co-rotational FEM approach with other models. On the first row are shown input RGB images, the second row features the corresponding segmented frame, the third row shows the 3D mesh tracking the object with the mass spring model, the fourth with the standard linear FEM

model, the fifth with the co-rotational model, and the last along with contour weighting. We can notice the ability of the process of the proposed method to correctly segment the visible part of the object, to track rigid motions and, in contrast to the mass spring and standard FEM models, to accurately register deformations, while being robust to occlusions due to the hands manipulating the object (third column on 7) or segmentation errors. The slight advantage of the contour weighting technique can be observed when stretching the object.

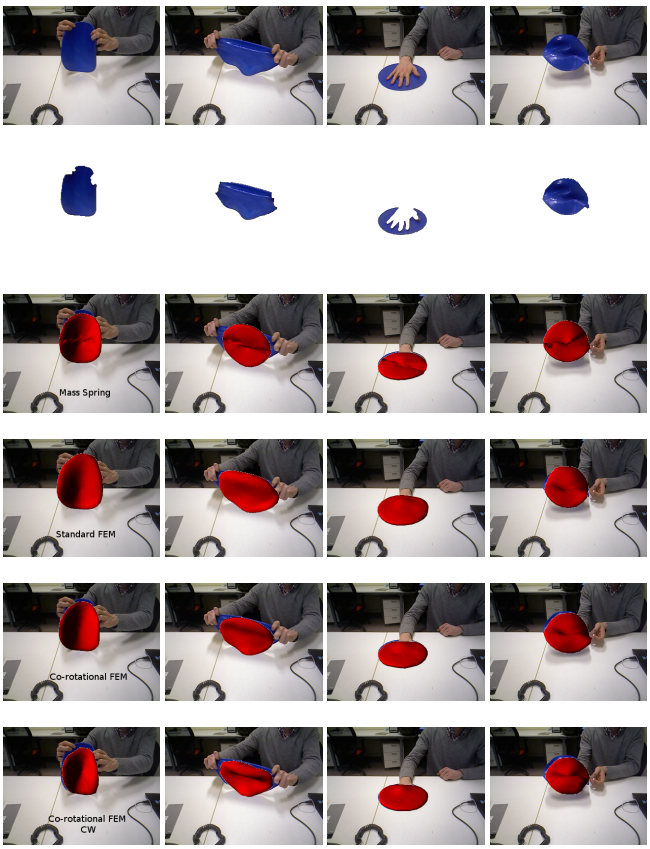

Fig. 7: Results of the tracking process for the pizza-like object, with the input images (first row), the segmented frames (second row), and the registered mesh reprojected in the input image, for the mass spring model (third row), the standard FEM model (fourth row) and finally with the co-rotational model, and with the contour weighting technique (CW).

With the cylindrical bar object, a sequence featuring rigid motions, along with bending deformations, is featured in Fig. 9. Here the circumferential/radial/height resolution of the mesh, depicted in Fig.8, is $10 \times 20 \times 2$, resulting in 420 vertices. The material is here poorly elastic, the Young Modulus being empirically set to $E = 0.900MPa$ and the Poisson ratio to $\nu = 0.3$. Satisfactory results are also observed regarding occlusions, tracking rigid motions and isometric deformations, in comparison with other methods (mass spring and standard FEM models).

1) Computational costs: Regarding computational aspects, in Tab. I are shown the computation times of the

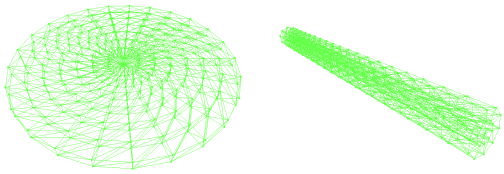

Fig. 8: Meshes used for the pizza-like object and the cylindrical bar object.

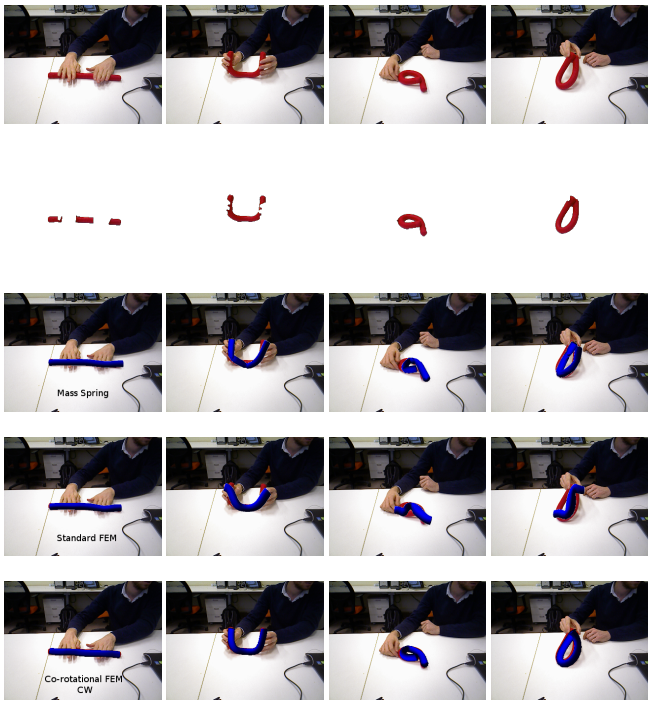

Fig. 9: Results of the tracking process for the cylindrical bar object, with the input images (first row), the segmented frames (second row), and the registered mesh reprojected in the input image, for the mass spring model (third row), the standard FEM model (fourth row) and finally with our approach.

various phases of the algorithm, for the different methods compared in this paper. *Visibility* corresponds to the process of determining the visible vertices of the rendered mesh, and in the case of using the contour weighting mode, extracting the vertices lying on the contour. *Ext. forces* is the step involving the determination of the closest points between the mesh and the point cloud, and the computation the subsequent external forces exerted on the mesh. *Resolution* consists in the resolution of the Lagrangian mechanical equations, to compute the deformations. The presented figures are the averages of the execution times per frame (in milliseconds) for the sequence presented in Fig. 7. As noticed, the suggested method (Co-rotational model with the contour weighting mode) runs on the sequence at around 35 fps. We can also observe that, the computational costs for the resolution phase being relatively small within the whole process, overall execution times are relatively independent of the selected model.

	Mass Spring	Stand. FEM	Corot.	Corot. - CW
Segmentation	10.7	10.5	10.7	10.7
Rigid ICP	3.0	2.5	2.7	2.6
Visibility	8.1	8.2	7.6	7.4
Ext. forces	3.4	3.5	3.5	4.0
Resolution	2.8	3.3	4.0	4.1
Total	28.0	27.9	28.6	28.8

TABLE I: Execution times, in milliseconds, for the different phases of the approach, and the various models and methods employed in this paper.

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 a textureless object subjected to various large deformations, with an 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. Regarding future works, efforts could be concentrated on different aspects to improve, such as segmentation, which could benefit from the depth data, the point cloud matching procedure, and the physical model, by extending it to other deformations such as plastic ones. A major issue would also consist in demonstrating the suitability of the approach to mimic the art of making pizzas with a dual arm/hand robot.

References

[1] Adrien Bartoli, Vincent Gay-Bellile, Umberto Castellani, Julien Peyras, Søren Olsen, and Patrick Sayd. Coarse-to-fine low-rank structure-from-motion. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

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

[3] 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.

[4] 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.

[5] Robert D Cook. *Finite element modeling for stress analysis*. Wiley, 1994.

[6] 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.

[7] 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.

[8] 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.

[9] Nazim Haouchine, Jérémie Dequidt, Marie-Odile Berger, Stéphane Cotin, et al. Single view augmentation of 3d elastic objects. In *International Symposium on Mixed and Augmented Reality-ISMAR*, 2014.

[10] 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.

[11] 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.

[12] 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.

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

[14] Vincenzo Lippiello, Fabio Ruggiero, and Bruno Siciliano. Floating visual grasp of unknown objects using an elastic reconstruction surface. In C. Pradalier, R. Siegwart, and G. Hirzinger, editors, *Robotics Research*, volume 70 of *Springer Tracts in Advanced Robotics*, pages 329–344. Springer Berlin Heidelberg, 2011.

[15] 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.

[16] 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.

[17] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1951–1958. IEEE, 2011.

[24] 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.