2016 Fourth International Conference on 3D Vision

Tracking deformable surfaces that undergo topological changes using an RGB-D camera

Aggeliki Tsoli, Antonis A. Argyros
Institute of Computer Science, FORTH, Greece
{aggeliki, argyros}@ics.forth.gr

Abstract

We present a method for 3D tracking of deformable surfaces with dynamic topology, for instance a paper that undergoes cutting or tearing. Existing template-based methods assume a template of fixed topology. Thus, they fail in tracking deformable objects that undergo topological changes. In our work, we employ a dynamic template (3D mesh) whose topology evolves based on the topological changes of the observed geometry. Our tracking framework deforms the defined template based on three types of constraints: (a) the surface of the template has to be registered to the 3D shape of the tracked surface, (b) the template deformation should respect feature (SIFT) correspondences between selected pairs of frames, and (c) the lengths of the template edges should be preserved. The latter constraint is relaxed when an edge is found to lie on a "geometric gap", that is, when a significant depth discontinuity is detected along this edge. The topology of the template is updated on the fly by removing overstretched edges that lie on a geometric gap. The proposed method has been evaluated quantitatively and qualitatively in both synthetic and real sequences of monocular RGB-D views of surfaces that undergo various types of topological changes. The obtained results show that our approach tracks effectively objects with evolving topology and outperforms state of the art methods in tracking accuracy.

1. Introduction

Tracking the deformations of real world shapes from visual input plays an important role in fields ranging from computer vision and medical imaging to robotics and computer graphics. There has been a lot of success in the past on tracking deformable objects of fixed topology [18, 8, 7]. However, tracking deformable surfaces with dynamically evolving topology is still an understudied problem with numerous applications. For instance, such a capability would provide very valuable input towards developing vision/robotic systems that can track and reason about activities such as tearing a paper, cutting a piece of bread, *etc*. As another example, tracking cutting and tearing may facilitate the development of relevant data-driven models or provide cues for animating the scene using existing elaborate fracture models developed in the graphics community.

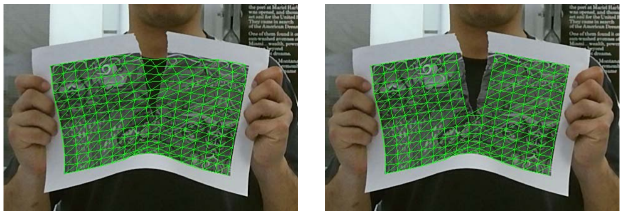

(a) fixed template (b) dynamic template

Figure 1: Tracking the deformations of a paper due to tearing. We initialize manually the location and geometry of the paper using a 2D grid (template) shown in green and show the registration when the paper is half-way torn using (a) a fixed template and (b) a dynamic template. We observe that the dynamic template matches more accurately the geometry of the paper over time.

Template-based methods have been used extensively for dense tracking of deformable objects with remarkable results. Previous work has demonstrated effective tracking of highly deformable surfaces such as faces [5], clothing [7], *etc*. These methods rely on the use of a fixed template, typically a 3D mesh, describing the shape of the object of interest and serving as a topological prior for tracking its deformations over time. However, the assumption of a topologically fixed template is not valid when the topology of the tracked object is dynamically evolving, e.g. in cases such as the gradual tearing of a paper, unzipping a jacket, *etc*. Figure 1a shows an example where a template with fixed topology is deformed unsuccessfully during the tearing of a sheet of paper. Having multiple potential templates is not practical because, i.e., in the case of tearing, we do not know beforehand where exactly on the surface the cut

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.42

will occur.

In this paper, we employ a template-based method for tracking, but we propose instead adapting the topology of the template based on the observed data. Figure 1b shows tracking the partial cut of a paper using our method. We assume that we know the initial shape of the object to be tracked (*template*), but have no prior knowledge about its shape deformations or material properties. Our input is monocular RGB-D data and the template is represented as a 3D mesh. For the first (reference) frame the template is manually registered to the observations. Moreover, a number of sparse texture features (SIFT) are automatically extracted and registered to the template. For each frame, our goal is to deform the template in a way that best achieves three goals. First, the template surface should be as close as possible to the surface of the observed 3D point cloud. Second, the deformation of the mesh should respect the correspondences of visual (SIFT) features detected in subsequent frames. To prevent from drifting during tracking, SIFT features are also matched between the current and the reference frame. Finally, in order to prohibit degenerate template deformations, the lengths of the template edges should not change considerably. We update the topology of the template by detecting edges that lie on a "geometric gap", i.e., edges along which a significant depth discontinuity exists. Overstretched edges that lie on a "geometric gap" are then removed from the template.

The resulting method allows for tracking both the non-rigid deformations of the surface of interest as well as its topological changes. While we assume almost inextensible surfaces, no other prior knowledge on the material properties of the surface to be tracked is required. Although the term "topological changes" refers to both cutting and merging geometries, we focus on cutting or tearing of a surface because (a) these topological changes are encountered more frequently in real life object manipulation scenarios, (b) merging is not always meaningful. For instance, when tearing a paper into 2 pieces, the torn mesh should not be stitched back when the sides of the cut come spatially close.

To the best of our knowledge, this is the first work that addresses tracking surfaces of dynamic topology due to complex tearing effects.

2. Related Work

We track highly deformable surfaces with dynamic topology from monocular RGBD input. There has been very limited work on tracking surfaces whose topology changes over time, e.g. objects that break, get torn etc. Therefore, we review methods that perform monocular tracking of deformable objects under the assumption of a fixed topology. Our method is also related to template-based tracking, 3D surface reconstruction, registration and 3D shape matching, so we briefly review relevant methods, regardless of the number of cameras they involve. Finally, we review a few works that are related to the tracking and modeling of surface cuts and tearing.

Monocular tracking of deformable objects: Recovering the shape of deformable surfaces from single images is inherently ambiguous [26], given that many different shape/camera configurations can produce the same images. The main approaches for deformable surface reconstruction based on 2D information either assume that a reference 2D template and corresponding 3D shape are known [20, 21, 1] or perform 2D tracking throughout a sequence of images [9, 8] (non-rigid Structure from Motion). As an example of the first class of methods, Ngo et al. [20] address the problem of 3D reconstruction of poorly textured, occluded surfaces, proposing a framework based on a template-matching approach that ranks dense robust features by a relevancy score. Ostlund et al. [21] track control points of a surface in 2D and infer its 3D shape using the control points and a Laplacian deformation model. Bartoli et al. [1] perform template-based deformable 3D reconstruction from a single input image and provide analytical solutions to the problem accounting for both isometric and conformal surface deformations. The works in [9, 8] present optical flow-based surface tracking. In [9] the optical flow field is regularized with a 2D mesh-based deformation model. The formulation of the deformation model contains weighted smoothing constraints defined locally on topological vertex neighborhoods. Garg et al. [8] exploit the high correlation between 2D trajectories of different points on the same non-rigid surface by assuming that the displacement of any point throughout the sequence can be expressed in a compact way as a linear combination of a low-rank motion basis. This subspace constraint acts, effectively, as a trajectory regularization term leading to temporally consistent optical flow.

When it comes to tracking from single view pointcloud data, Schulman et al. [27] proposed a real-time tracking algorithm based on a probabilistic generative model that incorporates observations of the point cloud and the physical properties of the tracked object and its environment. The algorithm is able to track robustly various types of deformable objects, including ones that are 1D such as ropes; 2D such as cloth; and 3D such as sponges. Tzionas et al. [29] track deformable objects in interaction with human hands, jointly. The hands and objects are represented as articulated meshes and their pose is inferred by fitting the meshes to the pointcloud data while ensuring physical plausibility by performing a physics-based simulation of the scene. Wuhrer et al. [31] combine a tracking-based approach with fitting a volumetric elastic model to improve the estimation of the unobserved side of an object from pointcloud data. Petit et al. [23] track in real-time a 3D object which undergoes large deformations such as elastic ones, and fast rigid motions. They perform non-rigid fitting of a mesh to the 3D pointcloud of the object based on the Finite Element Method to model elasticity and on geometrical point-to-point correspondences to compute external forces exerted on the mesh.

Template-based tracking: In template-based tracking methods, a template, typically a 3D mesh, serves as a topological prior for the object of interest. In addition, various shape deformation priors are built based on the template either in the form of a deformation model such as blendshapes [5], thin plate splines [15], NURBS [11] or using generic *smoothness constraints* to preserve the structure of the tracked object. Example smoothness constraints include enforcing similar transformations [13], velocity fields [18, 19] between nearby template vertices, or similar triangle transformations between neighboring triangles [10]. Existing methods for template-based tracking assume that the tracked template mesh is topologically fixed. Given that the smoothness constraints are defined based on the topology of the fixed template, they fail to capture surface deformations of objects whose topology changes. It has to be noted that dividing the object into parts and assigning a different template per part is not practical because neither the location of the cut nor the shapes of the resulting parts is known a priori. Letouzey et al. [12] present a method for inferring the optimal template for tracking. However, the inferred template is tailored to a specific sequence and does not adapt over time.

3D shape reconstruction, registration and matching: Recently a number of methods for the 3D reconstruction of dynamic geometries have been proposed [32, 3, 14, 19]. Contrary to these methods, our goal is to perform model-based tracking of such dynamic, topologically varying geometries and to provide dense point correspondences among all input frames. Our method can operate on top of such 3D reconstruction techniques.

Tam et al. [28] present an overview of previous work on registering 3D data. We distinguish the Coherent Point Drift method [18] for non-rigid registration as one of the most robust ones to changes in topology. This is due to the fact that the neighborhood of a vertex on the template is not defined based on edges, but rather using Gaussians with infinite support centered on the template vertices.

Most of the existing 3D shape matching approaches [30] typically provide sparse correspondences between the matched shapes. In contrast, our approach provides dense correspondences. Moreover, many of the proposed shape descriptors used for matching are based on the isometry assumption [4], according to which the geodesic distances between surface points are preserved. However, this assumption does not hold in surfaces of changing topology.

Tracking and modeling tearing: Fracture modeling has attracted the interest of the graphics community for a long time and very elaborate fracture models for objects of various materials have been proposed [6, 25]. Due to the low-resolution of our input data (especially the depth) as well as for computational efficiency, we employ weak generic deformation priors that resemble a spring-based model with stretching constraints similar to the ones in Position Based Dynamics [17, 2].

Recently, Paulus et al. [22] and Petit et al. [24] presented augmented-reality oriented methods for tracking the tearing of deformable objects. Contrary to our method, they assume knowledge of the material properties of the object to be tracked which is then used to built a physics-based model of the object. Setting the parameters for such a model is typically a cumbersome process that has to be repeated for each object (material) type. Moreover, having a physics-based representation makes it hard to track accurately complex tearing effects. For instance, [22] handles single cuts. In contrast, our approach is able to handle multiple, intersecting cuts.

Our contribution: To the best of our knowledge, we present the first method that is able to track accurately, highly deformable surfaces that undergo multiple, intersecting cuts based on RGBD input. We support our claim based on an extensive, quantitative evaluation of the proposed method that is performed on a new dataset[1] of synthetic sequences that is annotated with ground truth information. The employed surface templates consist of a large number $(\sim 1K)$ of vertices. We also compare the results we obtain with those of existing, state of the art methods [18, 8]. The comparison reveals that the proposed method performs comparably to the state of the art in tracking deformable objects of fixed topology, but clearly outperforms it for surfaces of evolving topology. Additionally, we provide qualitative results on real world sequences. These show that our method can track challenging topological changes of objects with very different physical properties (the opening of a door, the cutting of a loaf of bread and the double cutting of a deformable sheet of paper), all treated with the same algorithmic parameters.

3. The Proposed Method

Our input data is a monocular RGBD sequence $\{I_f, D_f\}_{f=1}^{K}$ consisting of K frames where I_f and D_f are the RGB image and corresponding depth map at frame f, respectively. To eliminate high-frequency noise on the depth values, we perform bilateral filtering on each depth map. Additionally, we treat points on the depth map with no measurements as if they belong to the background. We assume knowledge of the camera's projection matrix P and, based on this, we derive a point cloud P_f out of each depth map D_f.

We denote with $M_f = (V_f, \mathcal{E}_f)$ the template mesh at a frame f. The reference template consists of N vertices

[1] Publicly available at http://www.ics.forth.gr/cvrl/tearing/.

stored in $V_f = [\mathbf{v}_1^f \dots \mathbf{v}_N^f] \in \mathbb{R}^{3\times N}$ where each column represents a vertex. Thus, in general, $\mathbf{v}_i^j$ represents the i-th vertex of the mesh at frame j. Additionally, the connectivity of the template mesh is expressed through a set of edges $\mathcal{E}_f \subset V_f \times V_f$. We assume that for the first (reference) frame of the sequence the template is manually registered to the visual data, that is $M_1 = (V_1, \mathcal{E}_1)$ is known.

For each RGB frame I_f, we extract a set S_f of N_f SIFT features [16], $S_f = \{\mathbf{s}_i^f\}_{i=1}^{N_f}$. Given the registration of I_f with D_f and P_f, we assume that all SIFT features $\mathbf{s}_i^f$ are represented as 3D points in the camera centered coordinate system. Finally, we denote with $C_k(\mathbf{s}_j^f)$ the corresponding of feature $\mathbf{s}_j^f$ at frame k.

Our goal is then to infer $\{M_f = (V_f, \mathcal{E}_f)\}_{f=2}^K$, that is the 3D coordinates of the template vertices $\{V_f\}_{f=2}^K$ as well as the template connectivity expressed through the set of edges $\{\mathcal{E}_f\}_{f=2}^K$.

3.1. Rigid registration

At each frame f, we initially perform a rigid registration of the previous set of vertices V_{f-1} and the point cloud P_f based on the 3D coordinates of the SIFT feature matches between frames $f-1$ and f. In practice, this step reduces the number of optimization steps needed later to infer V_f at the current frame.

To do so, we estimate the rigid transformation T_f that connects the SIFT feature correspondences between frames $f-1$ and f. In this process, we do not take into account SIFT matches involving features at frame $f-1$ that are located further than $d = 1cm$, for real data, from the surface of the template M_{f-1}. Subsequently, the transformed nodes V_f are computed by applying the transformation matrix T_f to the nodes of V_{f-1}.

3.2. Non-rigid registration

As a next step, we perform non-rigid registration between V_f and the pointcloud P_f and get an updated estimate V_f' of V_f. We cast it as a minimization problem

$$V_f' = \mathrm{argmin}_{V_f}\, E(V_f, P_f, S_f, S_{f-1}, S_1, \mathcal{E}_f, V_1) \qquad (1)$$

of the following energy function:

$$\begin{aligned} E(V_f, P_f, S_f, S_{f-1}, S_1, \mathcal{E}_f, V_1) = \lambda_G E_G(V_f, P_f) \\ + \lambda_F E_F(V_f, S_f, S_{f-1}, S_1) \\ + \lambda_S E_S(V_f, \mathcal{E}_f, V_1). \end{aligned} \qquad (2)$$

3.2.1 Registering the geometry of the template to the point cloud

The first term in Eq.(2) aims at bringing the geometry of the template as close as possible to that of the point cloud. So, $E_G(V_f, P_f)$ is defined as:

$$E_G(V_f, P_f) = \sum_{i=1}^{N} ||\mathbf{v}_i^f - \mathbf{g}_i^f||_2^2, \qquad (3)$$

where $\mathbf{g}_i^f$ is the closest point of (the current estimate of) the template vertex $\mathbf{v}_i^f$ on the pointcloud P_f.

3.2.2 Accounting for feature correspondences

We, additionally, drive the fit of the template to the observed geometry by matching SIFT features between frames. For a certain frame f, this is performed relative to the previous frame $f-1$. However, to minimize drift, this is also performed relative to the reference frame $f = 1$. Typically, the SIFT matches with respect to the reference frame are much fewer than the matches with respect to the previous frame.

For each SIFT feature $\mathbf{s}_i^f$ we compute its projection $b_f(\mathbf{s}_i^f)$ on the surface of M_f. Essentially, this entails (a) finding the triangular patch of M_f on which $\mathbf{s}_i^f$ projects and (b) expressing $\mathbf{s}_i^f$ in barycentric coordinates. This way, a SIFT feature is expressed as a function of the coordinates of the vertices of the template which permits the deformation of the template.

Given the above, $E_F(V_f, S_f, S_{f-1}, S_1)$ is defined as:

$$\begin{aligned} E_F(V_f, S_f, S_{f-1}, S_1) = t_1 \sum_{j=1}^{p_f} \left|\left| b_{f-1}(\mathbf{s}_j^{f-1}) - c_f(\mathbf{s}_j^{f-1}) \right|\right|_2^2 \\ + t_2 \sum_{k=1}^{r_f} \left|\left| b_1(\mathbf{s}_k^1) - c_f(\mathbf{s}_k^1) \right|\right|_2^2 . \end{aligned} \qquad (4)$$

The first term in Eq.(4) accounts for the p_f feature correspondences between frames $f-1$ and f, while the second term accounts for the r_f feature correspondences between frames 1 and f. The scalars t_1, t_2 determine the relative importance of the features from the previous and reference frames and are set empirically to $t_1 = 1$, $t_2 = 2$.

3.2.3 Preserving structure

Finally, the third term in Eq.(2) aims at preserving the lengths of the edges of the template, as those were defined in M_1. Thus, $E_S(V_f, \mathcal{E}_f, V_1)$ is defined as:

$$E_S(V_f, \mathcal{E}_f, V_1) = \sum_{e_{ij} \in \mathcal{E}_f} w_{ij} \left(||\mathbf{v}_i^f - \mathbf{v}_j^f||_2 - ||\mathbf{v}_i^0 - \mathbf{v}_j^0||_2 \right)^2 \qquad (5)$$

where $e_{ij} = \{\mathbf{v}_i^f, \mathbf{v}_j^f\}$. In Eq.(5), w_{ij} is a scalar that weights the contribution of each edge to the error term. During this step all w_{ij} are set equal to $w = 1$. However, w_{ij}

assume different values when accommodating for topological changes of the template (see Sec. 3.3).

3.2.4 Optimization

At each frame f, the minization problem of Eq.(1) is solved based on the Levenberg-Marquardt method, initialized with the inferred coordinates of the template vertices at the previous frame $f-1$ after rigid registration (see Sec. 3.2). The weights λ_G, λ_F and λ_S weigh the relative importance of the corresponding error terms, were empirically set to $\lambda_G = 2$, $\lambda_F = 15$ and $\lambda_S = 7$ and were kept constant throughout all experiments. The optimization stops either when the average distance between the inferred 3D locations of the template vertices at two consecutive iterations is less than $1mm$ or when a maximum number of 20 iterations is reached.

3.3. Non-rigid registration handling topological changes

In the case of topological changes, the non-rigid registration step above will yield a not so meaningful fit to the data because the assumption of edge length preservation is no longer valid. To accommodate changes in topology, we repeat the previous step by adjusting the weights w_{ij} of edges in Eq.(5) based on whether they lie on a "geometry grap", that is whether there is a depth discontinuity on the depth map along a certain edge.

To determine the likelihood $y(i,j)$ that an edge $\{\mathbf{v}_i^f, \mathbf{v}_j^f\}$ lies on a geometric gap, we consider $L = 100$ 3D points $\{\mathbf{u}_l^{(i,j)}\}_{l=1}^L$ uniformly distributed along the edge in 3D. We project these 3D points on the depth map D_f and then we lift these projections to 3D points $\{\mathbf{d}_l^{(i,j)}\}_{l=1}^L$. Then, we define

$$y(i,j) = \frac{1}{L}\sum_{l=1}^{L} z_{ij}(||\mathbf{u}_l^{(i,j)} - \mathbf{d}_l^{(i,j)}||_2), \tag{6}$$

with

$$z_{ij}(x) = \begin{cases} 1, & \text{if } x > T_{ij} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In Eq.(7), we set $T_{ij} = ||\mathbf{v}_i^f - \mathbf{v}_j^f||$. Then, we update the weights w_{ij} as follows:

$$w_{ij} = we^{-cy(i,j)}. \tag{8}$$

In Eq.(8), we set $c = 6$. Minimizing the energy in Eq.(2) using V_f' for initialization and the weights in Eq.(8) results in the final estimate V_f'' of the template at the current frame f.

3.4. Template topology update

The last step of the method is to update the topological constraints of the template. Essentially, this amounts to removing template edges defined between nodes that should not be connected anymore due to topological changes of the surface. We update the template edges $\mathcal{E}_f$ by taking into account the template edges $\mathcal{E}_{f-1}$ of the previous frame $f-1$ and by removing edges based on two criteria: (a) whether an edge lies on a geometric gap and, (b) whether the topology prior is violated, that is an edge is overstretched. More specifically, if an edge lies on a geometric gap and is stretched more than T_s, as a ratio with respect to its initial size, it is removed. We set $T_s = 1.1$, that is an edge is considered overstretched when its current length is more than 10% larger than its initial length. In notation:

$$\begin{aligned} \mathcal{E}_f = \{\{\mathbf{v}_i^f, \mathbf{v}_j^f\} \mid \{\mathbf{v}_i^{f-1}, \mathbf{v}_j^{f-1}\} \in \mathcal{E}_{f-1} \wedge \\ y(i,j) > 0 \wedge \frac{||\mathbf{v}_i^f - \mathbf{v}_j^f||}{||\mathbf{v}_i^0 - \mathbf{v}_j^0||} > T_s\}. \end{aligned} \tag{9}$$

One might think that updating the topology of the template could potentially be performed in a trivial way, i.e., by using an existing non-rigid template registration method and by removing edges based on whether they lie on a geometric gap. This simple idea does not work in practice because the cut of the surface affects the overall fit of the template to the data (see Figure 1a). Thus, removing template edges in this naive way may lead to over-removal of edges.

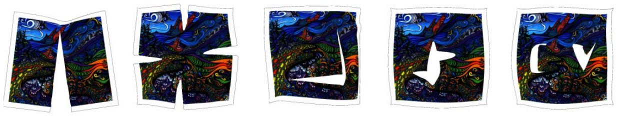


Figure 2: Synthetic data. We show the final frame per sequence where the cut of the surface is most pronounced. The ground truth template consists of $\sim 1K$ vertices and each sequence consists of 25-30 frames.

4. Experimental Results

Public datasets that showcase geometries that change topology over time are limited in the sense that topology changes occur mainly due to objects that come into contact and then separate again. We evaluate quantitatively our method using synthetic data generated through physics-based simulation in the 3D modeling software Blender (Figure 2) as well as qualitatively based on monocular RGBD data of surfaces of various materials captured with a Microsoft Kinect 2 (Figure 6).

4.1. Evaluation on synthetic data

We define two variants of our method. The "no cut" variant employs the template-based registration part (sections 3.1 and 3.2) without updating the template topology. The "cut" variant treats also topological changes (sections

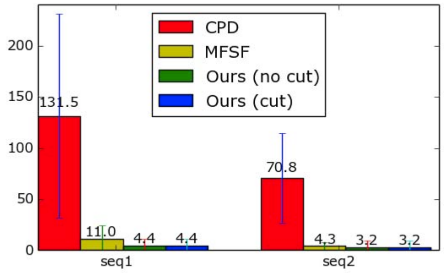


Figure 3: Evaluation on tracking deformable surfaces *with no change* in topology. We show the per-vertex Euclidean distance from ground truth for deformable surfaces with around 400 and 600 vertices. The distance is averaged over all vertices and over all 73 and 30 frames of the two sequences, respectively. Additionally, it is expressed as a percentage of the cell width (the largest side) of the underlying grid.

3.3 and 3.4). We compare both variants against CPD [18], a non-rigid registration method that considers only the observed geometry, and MFSF [8] that operates only on image (2D) data. For both CPD and MFSF we use the implementations available online[2]. Comparison with previous work on tracking cuts [22, 24] and monocular tracking of deformable surfaces from RGBD [27, 11] was not feasible due to unavailability of code and/or data.

4.1.1 Evaluation metrics

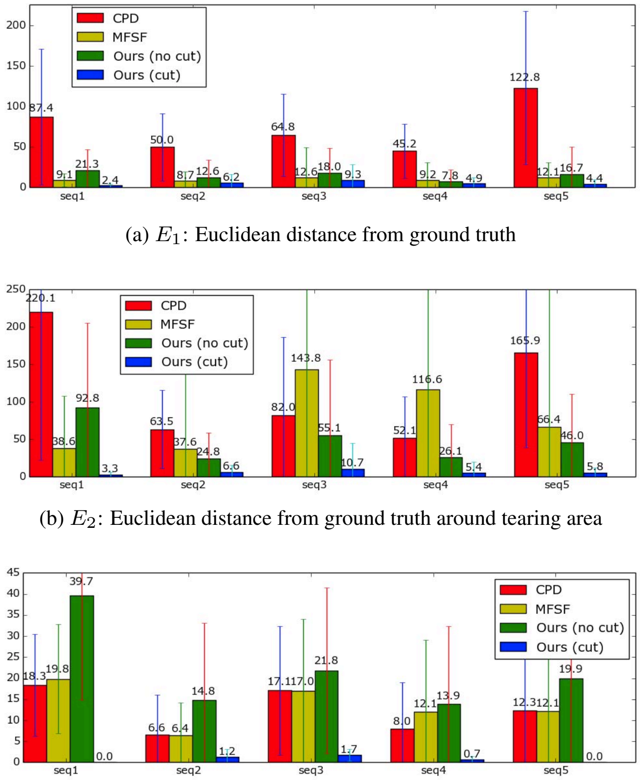
We evaluate the above mentioned methods using three error metrics. The first one, E_1, denotes the average Euclidean distance between the 3D locations of the template vertices as inferred by each method and their ground truth locations. Given that [8] operates only in 2D, we calculate the 3D locations of the template vertices on the observed pointcloud using the camera's projection matrix. In this case, however, vertices that map to the "background" of the depth map can be arbitrarily far from the pointcloud of the deformable surface. To make the comparison among methods as fair as possible, we consider only the vertices $\mathbf{v}_i^f$, $i \in N_f^*$, that are "on surface" at each frame f and we do that for all methods. Thus,

$$E_1 = \sum_{f=1}^{K} \sum_{i \in N_f^*} ||\mathbf{v}_i^f - \mathbf{x}_i^f||_2, \tag{10}$$

where $\mathbf{x}_i^f$ is the ground truth location of vertex $\mathbf{v}_i^f$.

To highlight the effect of topological changes on the registration, we evaluate the same metric but only on a neighborhood around the topological change. This neighborhood, N_f^{**}, comprises of all vertices adjacent to the ground truth torn edges as well as the vertices one edge away. Thus,

$$E_2 = \sum_{f=1}^{K} \sum_{i \in N_f^{**}} ||\mathbf{v}_i^f - \mathbf{x}_i^f||_2. \tag{11}$$



(a) E_1: Euclidean distance from ground truth

(b) E_2: Euclidean distance from ground truth around tearing area

(c) E_3: Number of vertices off the surface

Figure 4: Evaluation on tracking deformable surfaces *with change* in topology (see text for details).

Ideally, no template vertex should appear off the surface of the observed pointcloud. Therefore, the third metric E_3 shows the number of such vertices:

$$E_3 = \sum_{f=1}^{K} |O^f|, \quad O^f = \{\mathbf{v}_i^f \mid ||\mathbf{v}_i^f - \mathbf{g}_i^f||_2 > T_b\}. \tag{12}$$

In Eq.(12), vertices are included in the set O_f if their distance to the closest point on the point cloud, $\mathbf{g}_i^f$, is larger than a predetermined threshold $T_b = 1cm$.

4.1.2 Deformable tracking *with no* topological change

Out of the four methods mentioned above, only our method is intentionally designed to account for topological changes. To establish a baseline performance per method as well as provide a comparison among them that is not influenced by topological changes, we assess their performance on two sequences of deforming surfaces *with no* topological changes.

[2] MFSF: https://bitbucket.org/troussos/mfsf, CPD: https://sites.google.com/site/myronenko/research/cpd

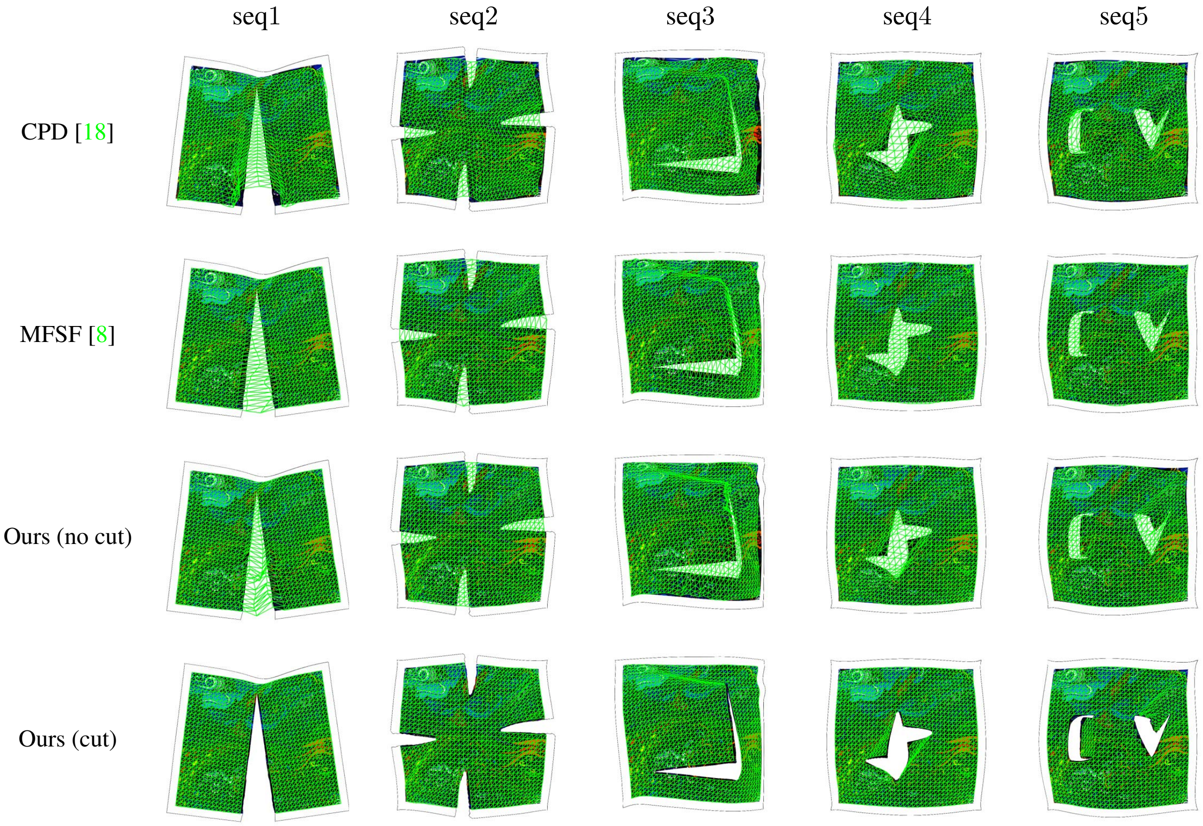


Figure 5: Visual evaluation of the performance of CPD [18], MFSF [8], our method using a fixed template (no cut) and our method using a dynamic template (cut) for the last frame of each of the five synthetic sequences that exhibit changes in topology. We observe that our method which accounts for topological changes provides more meaningful registration than the rest of the methods.

The sequences were generated by animating a grid with approximately 400 and 600 vertices respectively. Figure 3 shows the per vertex distance from ground truth (E_1 metric, see Sec. 4.1.1), averaged over all frames of the two sequences, along with the standard deviation of these distances. The distance is expressed as a percentage of the largest edge of the template. We observe that all methods are quite effective at tracking deformable surfaces with fixed topology. The higher error for CPD is justified by the fact that it relies solely on geometry for registration which may lead to "sliding" of the template along the observed surface. Our method is more accurate than the rest. We also observe that both variants of our method ("cut" and "no cut") perform equally well, which shows the robustness of the "cut" method in the case where no topological changes are present.

4.1.3 Deformable tracking *with* topological change

We evaluate the four methods mentioned above using the five synthetic sequences in Figure 2 displaying a variety of topological changes. Starting with a single cut (seq1), we progress to multiple non intersecting cuts (seq2), two or more intersecting cuts (seq3, seq4), multiple internal cuts of various shapes (seq 5). The surfaces of all sequences are also subject to small elastic deformations. In each sequence, the ground truth surface is represented as a grid of $\sim 1K$ equally spaced vertices. Each sequence contains 25-30 frames.

Figure 4 displays the performance of each method based on the three presented error metrics E_1, E_2 and E_3. Our method (cut) outperforms all other methods that do not account for topological changes both in terms of the distance from ground truth as well as the number of template vertices that end up off the surface of the observed geometry. The difference in performance is more pronounced when considering a neighborhood around the tearing area. This is also evident in Figure 5 that provides a visual evaluation of all methods for the last frame of each sequence. Our online video (`https://youtu.be/Hxa7nKUvsso`) provides full tracking results of the abovementioned methods for all synthetic sequences.

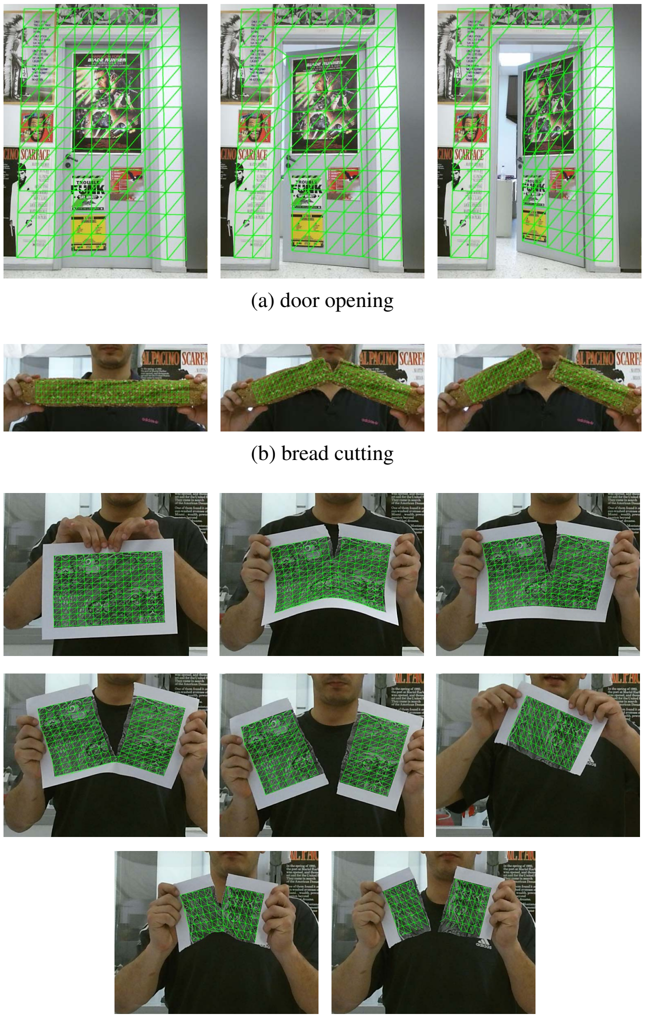
4.2. Evaluation on real data

We, additionally, evaluate our method on sequences of real objects and surfaces captured with a Microsoft Kinect 2.

Figure 6a shows an example of a 3D scene evolution. Starting with a 3D mesh of a scene, a door and its surrounding wall, the opening of the door causes topological changes in the observed geometry which are reflected on the topology of the scene mesh. Note that our method is tolerant to non-frontal views as long as there is no self-occlusion.

The scene described above is piecewise rigid. Our method has been applied successfully to tracking surfaces that are cut in a less principled way as well as highly deformable surfaces. Figure 6b shows tracking the surface of a loaf of bread cut into two pieces. Although it happens that the template is cut along a single column of the template grid, the underlying geometry around the tearing area could not have been produced e.g. using a cutting plane. We also observe that the initial (reference) template of the surface does not need to be planar.

Figure 6c shows tracking the deformations of a paper and the change in its topology during tearing. The template matches effectively both the shape of the paper around the tearing area as well as the deformations of the paper on the rest of its surface. The ability of our method to adapt the topology of the template during tracking and provide global correspondences over all frames in a sequence is more prominent when the paper is torn twice, consecutively. For a more detailed view of the tracking results, see https://youtu.be/Hxa7nKUvsso.



(a) door opening

(b) bread cutting

(c) paper tearing

Figure 6: Visual evaluation of our method on real data obtained with a Microsoft Kinect 2. (a) Tracking the evolution of a scene comprising of a door and its surrounding walls, (b) cutting a loaf of bread, (c) tearing a paper twice. Our method is able to track successfully multiple topological changes on surfaces that undergo large shape deformations such as in (c). It is also tolerant to non-frontal views as in (a) as long as there is no self-occlusion. Although we assume knowledge of the template mesh at the reference frame, the template does not need to be planar (a,b).

5. Conclusions and Discussion

We presented the first method that is able to track accurately highly deformable surfaces that undergo multiple, intersecting cuts based on RGBD input. Our method assumes knowledge of the initial shape and position of the object of interest, but has no prior knowledge about the type of shape deformations and topological evolutions that will be observed. In order to deal with the dynamic topology of deformable surfaces we proposed tracking using a template with dynamic topology. The edges of the template are constrained to retain their initial lengths; this constraint is relaxed when an edge lies on a geometric gap to allow for potential edge stretching in order to match the underlying geometry. We update the template by removing edges that lie on a "geometric gap" and get overstretched. Experimental results demonstrated the effectiveness of the proposed method and its improved accuracy compared to the current state of the art.

In our work, we focus on objects represented as 2D surfaces. We also assume that we start with full knowledge of the shape of the object provided by a template mesh and the object is always visible. That is, we do not consider the scenario of fusing geometry as we see different sides of the object. Future work will explore extending the method to tracking 3D objects from monocular or multiview data towards raising these limiting assumptions.

Acknowledgements. This work was partially supported by projects FP7-ICT-2011-9 WEARHAP and FP7-IP-288533 Robohow. The authors thank Konstantinos Roditakis for assistance with data acquisition.

References

[1] A. Bartoli, Y. Gerard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-template. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(10):2099–2118, 2015. 2

[2] J. Bender, M. Müller, and M. Macklin. Position-based simulation methods in computer graphics. *EUROGRAPHICS Tutorial Notes*, 2015. 3

[3] M. Bojsen-Hansen, H. Li, and C. Wojtan. Tracking surfaces with evolving topology. *ACM Transactions on Graphics (TOG)*, 31(4):53–1, 2012. 3

[4] D. Boscaini, J. Masci, E. Rodola, M. M. Bronstein, and D. Cremers. Anisotropic diffusion descriptors. *Computer Graphics Forum - Proc. EUROGRAPHICS*, 35(2), 2016. 3

[5] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40, 2013. 1, 3

[6] O. Busaryev, T. Dey, and H. Wang. Adaptive fracture simulation of multi-layered thin plates. *ACM Transactions on Graphics (TOG)*, 32(4):52:1–52:6, 2013. 3

[7] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1753, 2009. 1

[8] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 104(3):286–314, 2013. 1, 2, 3, 6, 7

[9] A. Hilsmann and P. Eisert. Tracking deformable surfaces with optical flow in the presence of self occlusion in monocular image sequences. In *Computer Vision and Pattern Recognition Workshops*, pages 6, 1, 2008. 2

[10] D. A. Hirshberg, M. Loper, E. Rachlin, and M. J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *European Conference on Computer Vision (ECCV)*, pages 242–255. Springer, 2012. 3

[11] A. Jordt and R. Koch. Fast tracking of deformable objects in depth and colour video. In *BMVC*, pages 1–11, 2011. 3, 6

[12] A. Letouzey and E. Boyer. Progressive shape models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 190–197, Providence, USA, June 2012. 3

[13] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia)*, 28(5), December 2009. 3

[14] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, and S. Rusinkiewicz. Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics (TOG)*, 31(1), January 2012. 3

[15] J. Lim and M.-H. Yang. A direct method for modeling nonrigid motion with thin plate spline. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1196–1202, 2005. 3

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004. 4

[17] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007. 3

[18] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010. 1, 3, 6, 7

[19] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352, Boston, MA, USA, June 2015. 3

[20] D. T. Ngo, S. Park, A. Jorstad, A. Crivellaro, C. Yoo, and P. Fua. Handling occlusions and sparse textures in a deformable surface tracking framework. *CoRR*, abs/1503.03429, 2015. 2

[21] J. O. M. Östlund, A. Varol, T. D. Ngo, and P. Fua. Laplacian meshes for monocular 3D shape recovery. In *European Conference on Computer Vision (ECCV)*, 2012. 2

[22] C. J. Paulus, N. Haouchine, D. Cazier, and S. Cotin. Augmented reality during cutting and tearing of deformable objects. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 54–59. IEEE, 2015. 3, 6

[23] A. Petit, V. Lippiello, and B. Siciliano. Tracking an elastic object with an RGB-D sensor for a pizza chef robot. 2

[24] A. Petit, V. Lippiello, and B. Siciliano. Tracking fractures of deformable objects in real-time with an rgb-d sensor. In *IEEE International Conference on 3D Vision (3DV)*, pages 632–639, 2015. 3, 6

[25] T. Pfaff, R. Narain, J. M. de Joya, and J. F. O'Brien. Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics (TOG)*, 33(4):xx:1–9, 2014. 3

[26] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2

[27] J. Schulman, A. Lee, J. Ho, and P. Abbeel. Tracking deformable objects with point clouds. In *International Conference on Robotics and Automation (ICRA)*, 2013. 2, 6

[28] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013. 3

[29] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *arXiv preprint arXiv:1506.02178*, 2015. 2

[30] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707, 2011. 3

[31] S. Wuhrer, J. Lang, and C. Shu. Tracking complete deformable objects with finite elements. In *3DIMPVT*, pages 1–8, 2012. 2

[32] A. Zaharescu, E. Boyer, and R. Horaud. Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):823–837, 2011. 3