Joint 3D Tracking of a Deformable Object in Interaction with a Hand

Aggeliki Tsoli and Antonis A. Argyros

Institute of Computer Science, FORTH, Greece
{aggeliki, argyros}@ics.forth.gr

Abstract. We present a novel method that is able to track a complex deformable object in interaction with a hand. This is achieved by formulating and solving an optimization problem that jointly considers the hand, the deformable object and the hand/object contact points. The optimization evaluates several hand/object contact configuration hypotheses and adopts the one that results in the best fit of the object's model to the available RGBD observations in the vicinity of the hand. Thus, the hand is not treated as a distractor that occludes parts of the deformable object, but as a source of valuable information. Experimental results on a dataset that has been developed specifically for this new problem illustrate the superior performance of the proposed approach against relevant, state of the art solutions.

1 Introduction

Deformable objects are nearly everywhere and humans interact with them continuously. Thus, tracking the interaction of human hands with such objects based on visual input can support a number of applications in domains that include but are not limited to augmented/virtual reality and robotics.

Although there has been a lot of previous work on capturing the interaction of hands with rigid objects [8,20,2,10,11,34], there has been very limited work on capturing the interaction of hands with complex deformable objects under arbitrary contact configurations. The most related work so far is the one by Tzionas et al. [42] on hands and articulated object tracking based on depth input. To capture deformations, this approach models non-rigid objects as articulated objects with a tree-structured skeleton and a large number of bones. When the fingers of a hand come close to the object, the tracking of the hand and the object is aided by physics-based modeling and simulation (Bullet). This works fine, unless the object exhibits complex deformations that cannot be modeled effectively by the assumed tree-structured, skeleton-based representation (e.g., general deformation of a planar sheet of paper or of a piece of cloth). Furthermore, tracking relies solely on depth information, so fitting the template mesh of the object to the observed point cloud becomes susceptible to sliding.

To deal with these problems, we propose an approach where the deformable object is modeled as a 3D triangular textured mesh. The method reasons explicitly and in every frame about the contact configuration of all fingertips with the

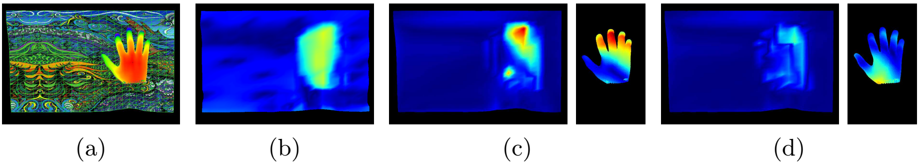


Fig. 1: Contrary to previous work on tracking hands in interaction with non-rigid objects, we capture complex object deformations by modeling the object with a textured 3D mesh. (a) A result of joint optimization for a frame. Colors on the hand show points that are closer (red) or further away (blue). (b) Heatmap-based visualization of object tracking error with [41]. (c) Heatmap-based visualization of the error in object tracking (left) and hand tracking (right) when performed independently. (d) Same as (c), but for the proposed joint optimization method.

object. The proposed joint tracking framework (see Fig. 1) relies on concepts from the Shape from Template community to model and track the object, and on a hybrid approach for tracking the hand. An initial estimate of the pose of the hand is computed as in [24], i.e., by fitting a 3D hand model to the 2D joint locations estimated by the discriminative method in [32]. Given this initial estimate, we can compute the area of the object that is occluded by the hand. Then, we refine the 3D hand pose and estimate the 3D deformation of the object jointly. This is formulated as an optimization problem that minimizes an error function that considers a variety of image features between consecutive frames as well as between the reference and the current frame, while keeping the mesh of the object close to the observed data. We, additionally, preserve the overall structure of the mesh using inextensibility constraints and a Laplacian-based smoothness prior [36]. Still, these constraints are not sufficient to represent the typically non-smooth deformations of the occluded part of the object occurring due to contact with the hand. To deal with this problem, our method reasons about contacts explicitly. More specifically, in our joint optimization framework, we examine all $2^5 = 32$ possible contact/no-contact configurations of the 5 fingertips with the object and we select the one that yields the best fit of the object's model to the observed data in the area around the hand. Essentially, the best contact configuration is the one that maximizes the fit of the deformable object model to the available observations, after tracking jointly the hand and the object.

The proposed approach is the first to achieve detailed tracking of complex deformable objects that are represented as 3D meshes and which interact with a human hand in a complex way. We show that the tracking of the deformable object is performed with an accuracy that has not been achieved before. At the same time, by plugging it in our joint optimization framework, the performance of a state of the art hand tracker is considerably improved. Our approach can be used in grasping scenarios with various contact configurations and relative distances between the object and hand and it is particularly effective in fine manipulation scenarios, i.e., finger tapping on the object of interest. We evaluate our method quantitatively on a synthetic dataset[1] we developed for this purpose. We also compare our approach to a state of the art RGBD-based method for deformable surfaces tracking [41] and we provide results obtained from a state of the art RGB-based method [17] as a reference baseline. Additionally, we showcase our method on real data and on a variety of object deformations produced by the interaction of an actual hand with materials such as cloth, paper and carton.

2 Previous Work

We present previous work on tracking deformable objects, hands as well as hand and object interactions from monocular input.

Monocular tracking of deformable objects: Recovering the shape of deformable surfaces from single images is inherently ambiguous [29], given that many different shape/camera configurations can produce the same images. Shape-from-Template methods reconstruct a deformable surface assuming that a reference 2D template and the corresponding 3D shape of the object are known. For instance, Ostlund et al. [22] 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. [3] 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 [9,6] present optical flow-based surface tracking. Parashar et al. [25] present volumetric Shape-from-Template to reconstruct the surface and interior deformation of a 3D object using constraints of local rigidity. Ngo et al. [17] 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. Their method is capable of considering an externally provided occlusion mask. This makes it possible to test this method in sequences with hand object interaction, i.e., by providing the hand mask that is computed automatically based on hand tracking. A direct comparison of this method to ours would be unffair (RGB vs RGBD input). However, we test this method on our datasets and provide the obtained results as a reference baseline.

With respect to tracking from single view point cloud data, Schulman et al. [30] propose an algorithm based on a probabilistic generative model that incorporates point cloud observations and the physical properties of the tracked object and its environment. Wuhrer et al. [47] combine a tracking-based approach with fitting a volumetric elastic model. Petit et al. [26] track a 3D object which undergoes large elastic deformations and fast rigid motions. They perform non-rigid fitting of a mesh to the 3D point cloud 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. Tsoli and Argyros [41] present a method for tracking isometric deformable surfaces that undergo topological changes such as a paper that is getting torn. We provide a comparison of our approach to this method.

[1] Available online at: https://www.ics.forth.gr/cvrl/deformable_interaction/.

Tracking of hands: A common categorization of hand motion/pose estimation methods is into generative, discriminative, and hybrid methods. Generative methods use a kinematics and potentially an appearance model of the hand to synthesize visual features that are comparable to the observed ones. Given that, a top-down search identifies the pose that maximizes the agreement between the synthesized and the observed visual features [27,19,10,11,37]. Generative methods result in accurate and physically plausible hand poses. On the other hand, their execution time and their inability to perform single-shot pose estimation are among their weak points. Discriminative methods for monocular hand tracking search in a large database of poses [28] or learn a mapping from depth and/or color input to pose space, typically in a large offline step [32,45,38,40,46,7,33,18]. Contrary to generative methods, discriminative ones are faster and do not suffer from drift, but may exhibit limited accuracy or the resulting estimation can be physically implausible. Hybrid approaches [35,44,31,48,16,24] combine the strengths of generative and discriminative ones. In our framework, we incorporate an extension of the RGB-based method in [24] that exploits RGBD data.

Tracking of hand-object interactions: Several previous works consider tracking hands in interaction with one or more known rigid objects [8,20,2,10,11,34]. In a few cases the models of rigid [23] or articulated [43] objects are reconstructed on the fly. With the exception of methods tracking two interacting hands (e.g., [21,11]) which therefore constitute special cases, Tzionas et al. [42] presented the single existing method for tracking a hand in interaction with an articulated object, from monocular RGBD. The hand and the object are represented as articulated meshes and their pose is inferred by fitting the meshes to the point cloud data while ensuring physical plausibility through a physics-based simulation of the scene. A problem with purely depth-based methods for tracking is that it is very hard to prevent the surface of interest (object or even the hand) from "sliding" along the surface of the observed point cloud. Moreover, the assumption of a tree-structured skeletal representation of the object limits the complexity of the deformations that can be handled effectively. On the contrary, we leverage on representing the non-rigid object as a 3D mesh and on using appearance features (SIFT, GBDF texture features) and inextensibility constraints to capture detailed mesh deformations. Deformable object tracking is also aided by considering it together with 3D hand pose estimation and by reasoning about hand-object contact points in a joint optimization framework.

Our contribution: We present the first method for tracking in 3D, the interaction of a human hand and a non-rigid object undergoing complex deformations from RGBD input. Our approach reasons about the contacts of a hand and the object during object manipulation. This is achieved through the joint optimization of the hand pose, the deformable object shape and their contact configuration that results in the best fitting of the deformable object model to the available visual observations in the vicinity of the detected and tracked hand. Our approach outperforms relevant existing methods on the first dataset that is suitable for the problem and which we will make publicly available.

3 The Proposed Method

Input: The input 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 the corresponding depth map at frame f. To eliminate high-frequency noise on the depth values, we perform bilateral filtering. We assume knowledge of the camera's projection matrix P and, based on this, we derive a point cloud P_f out of each D_f.

Deformable object model: We denote with $M_f = (V_f, \mathcal{E})$ the template mesh at a frame f. M_f consists of N_o vertices stored in $V_f = [\mathbf{v}_1^f \dots \mathbf{v}_{N_o}^f] \in \mathbb{R}^{3\times N_o}$ where each column represents a vertex and V_0 denotes the 3D vertex locations at the reference frame. Thus, $\mathbf{v}_i^j$ represents the i-th vertex of the mesh at frame j. The connectivity of the template mesh is expressed through a set of edges $\mathcal{E} \subset V_0 \times V_0$. We assume that for the first (reference) frame of the sequence the template is manually registered to the visual data, that is $M_0 = (V_0, \mathcal{E})$ is known and that the mesh topology $\mathcal{E}$ does not change over time.

Hand model: We use a hand model comprising of $B = 16$ bones and a detailed 3D triangular mesh with $N_h = 1597$ vertices. It has 26 degrees of freedom (DoFs) represented using 27 parameters. 7 are used to model the global translation and rotation (as quaternion) of the hand. The joint at the base of each finger is modeled using two DoFs and the rest of the finger joints require one DoF, each. The finger joints are bound by the joint limits that apply to a real hand [1]. Let $H(\theta_0) = (W_0, \theta_0)$ denote the 3D model of the hand at the reference pose θ_0 where $W_0 = [\mathbf{w}_1^0 \dots \mathbf{w}_N^0] \in \mathbb{R}^{3\times N}$ are the 3D locations of the surface vertices of the hand at the reference pose. Let also θ_f be the hand pose at frame f. The posed hand for the new pose θ_f is given by $H(\theta_f) = (W_f, \theta_f; W_0, \theta_0)$ where $W_f = [\mathbf{w}_1^f \dots \mathbf{w}_{N_h}^f] \in \mathbb{R}^{3\times N_h}$ denotes the vertices $\mathbf{w}_i^f$ on the surface of the hand that were transformed using linear blend skinning as:

$$\mathbf{w}_i^f = \sum_{b=1}^{B} a_{ib} T_b(\theta_f) T_b(\theta_0)^{-1} \mathbf{w}_i^0. \tag{1}$$

In Eq.(1), a_{ib} is the skinning weight of vertex i with respect to bone b and $T_b(\theta)$ denotes the global translation and/or rotation transformation of bone b due to pose θ. We define joints at the heads of all bones as well as at the tails of the bones closest to the fingertips and we end up with 21 joints in total. Let l_i^0 be the 3D location of each joint at the reference 3D model of the hand. The location of the joints at the posed hand model at frame f is given by

$$\mathbf{l}_i^f(\theta_f) = T_{b(i)}(\theta_f) T_{b(i)}(\theta_0)^{-1} \mathbf{l}_i^0, \tag{2}$$

where $b(i)$ is the bone that is associated with joint i. Our hand model was rigged in Blender [39]. The skinning weights are fixed for all frames and at most three skinning weights per vertex are nonzero.

Output: Our goal is to infer $\{M_f = (V_f, \mathcal{E})\}_{f=1}^{K}$, that is, the 3D coordinates of the template vertices $\{V_f\}_{f=1}^{K}$, as well as the pose of the hand $\{\theta_f\}_{f=1}^{K}$ at all frames f. This is performed in four steps. First, we estimate only the 3D pose of the hand from the input RGB image (Section 3.1). This gives us a rough estimate of the area where the hand lies, thus the area on the image where we expect the deformable object to be occluded. In turn, this provides an initialization for the second step in which we optimize jointly for the 3D location of the deformable object vertices and the pose of the hand (Section 3.2). Third, we select the optimal contact configuration of the fingers with respect to the deformable object (Section 3.3). Fourth, we fine-tune the joint fitting of the hand and the deformable object to our data considering the optimal contact configuration and coarse-to-fine texture features (Section 3.4).

3.1 Initial hand pose estimation

Given an RGB frame I_f and a bounding box around the hand, we estimate the 2D joint locations of the hand using the work in [32]. Let $J_i^f = (u_i^f, v_i^f, p_i^f)$, $i \in [1, 21]$, represent the 21 detected 2D hand joints at frame f. (u_i^f, v_i^f) are the 2D coordinates of the i-th joint on the input image I_f and p_i^f is the method's confidence for the joint i, ($p_i^f \in [0, 1]$). Let also $Q_i(\theta, P) = (x_i, y_i)$ be the projection of joints $\mathbf{l}_i(\theta)$ on the image plane, given (a) a pose θ and (b) the camera's projection matrix P. To avoid using false detections, we do not consider joints with confidence p_i below an experimentally identified value $p^{th} = 0.1$.

For a given pose θ, we quantify the discrepancy $d(Q_i(\theta, P), J_i)$ between the observed joint J_i and the computed one Q_i as in [24]:

$$d\left(Q_i(\theta, P), J_i\right) = (p_i^3 \cdot (x_i - u_i))^2 + (p_i^3 \cdot (y_i - v_i))^2. \tag{3}$$

Similarly, the total discrepancy between the observed and model joints is:

$$E_J(\theta) = \sum_{i=0}^{21} d(Q_i(\theta, P), J_i). \tag{4}$$

The 3D hand pose θ'_f that is most compatible to the observed 2D joints can be estimated by minimizing the objective function of Eq.(4):

$$\theta'_f = \arg\min_{\theta}\{E_J(\theta)\}. \tag{5}$$

This is achieved by the Levenberg-Marquardt optimizer [12,14] that minimizes this objective function after the automatic differentiation of the residuals. The bounding box around the hand is defined manually for the first frame and around the previous solution for the following frames.

The initial fitting of the hand provides a coarse occlusion mask $\mathcal{M}_f$ around the hand that we use both for fitting the deformable object (Section 3.2) and for assessing the quality of hypotheses about the contact configuration, i.e. which fingers touch the object (Section 3.3). The occlusion mask is calculated as the convex hull of the hand rendered at pose θ'_f dilated by a 50×50 kernel.

3.2 Joint estimation of hand pose and object deformation

We jointly estimate the pose θ_f of the hand and perform non-rigid registration of V_f on the point cloud P_f. This is performed by the minimization

$$V_f*, \theta_f* = \text{argmin}_{V_f, \theta_f} E(V_f, \theta_f, W_f, P_f, S_f, S_0, \mathcal{M}_f, V_0, \mathcal{E}, A, \mathbf{f}, Y), \tag{6}$$

of the following energy function:

$$\begin{aligned} E(V_f, \theta_f, W_f, P_f, S_f, S_0, \mathcal{M}_f, V_0, \mathcal{E}, A, \mathbf{f}, Y) &= \lambda_J E_J(\theta_f) + \ \lambda_{G\_h} E_G^h(W_f, P_f) \\ + \lambda_{G\_o} E_G^o(V_f, P_f) + \ \lambda_F E_F(V_f, S_f, S_{f-1}, S_0, \mathcal{M}_f) &+ \ \lambda_T E_T(V_f, V_0, \mathcal{M}_f) \\ + \lambda_S E_S(V_f, \mathcal{E}, V_0) + \ \lambda_L E_L(V_f, A) &+ \ \lambda_C E_C(V_f, \theta_f, W_f, \mathbf{f}, Y). \end{aligned} \tag{7}$$

In addition to the hand pose estimation error term E_J (see Section 3.1), the defined energy function consists of several error terms presented in detail below.

Registration of the geometry of the hand to the point cloud: The second term in Eq.(7) aims at bringing the visible geometry of the hand as close as possible to that of the point cloud. So, $E_G^h(W_f, P_f)$ is defined as:

$$E_G^h(W_f, P_f) = \sum_{\mathbf{w}_i^f \in \mathcal{W}_f} ||\mathbf{w}_i^f - \mathbf{g}_i^f||_2^2, \tag{8}$$

where $\mathcal{W}_f$ is the set of visible hand vertices $\mathbf{w}_i^f$ based on pose θ'_f and $\mathbf{g}_i^f$ is the closest point of $\mathbf{w}_i^f$ to the point cloud P_f.

Registration of the geometry of the template to the point cloud: In a similar way, the third term of Eq.(7) is designed to register the geometry of the deformable template to the point cloud. So, $E_G^o(V_f, P_f)$ is defined as the sum of distances of the template vertices $\mathbf{v}_i^f$ to their closest points $\mathbf{g}_i^f$ on the point cloud:

$$E_G^o(V_f, P_f) = \sum_{i=1}^{N_o} ||\mathbf{v}_i^f - \mathbf{g}_i^f||_2^2. \tag{9}$$

Account for feature correspondences: For each RGB frame I_f, we extract a set S_f of N_f SIFT features [13], $S_f = \{\mathbf{s}_i^f\}_{i=1}^{N_f}$. To make sure that we take into account SIFT features solely on the deformable object, we consider only features that are outside the hand mask $\mathcal{M}_f$ estimated in Section 3.1. 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 $c_k(\mathbf{s}_i^f)$ the corresponding of feature $\mathbf{s}_i^f$ at frame k. 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, $E_F(V_f, S_f, S_{f-1}, S_0, \mathcal{M}_f)$ is defined as:

$$E_F(V_f, S_f, S_{f-1}, S_0, \mathcal{M}_f) = t_1 \sum_{j=1}^{r_f^{f-1}} \left|\left|b_{f-1}(\mathbf{s}_j^{f-1}) - c_f(\mathbf{s}_j^{f-1})\right|\right|_2^2 + t_2 \sum_{i=1}^{r_f^0} \left|\left|b_1(\mathbf{s}_i^0) - c_f(\mathbf{s}_i^0)\right|\right|_2^2 , \tag{10}$$

for all r_f^0 features from the reference frame and r_f^{f-1} features from frame $f-1$ whose correspondences at frame f fall outside the hand mask $\mathcal{M}_f$. The scalars t_1, t_2 determine the relative importance of the features from the previous and reference frames and are set empirically to $t_1 = 1$, $t_2 = 5$.

Account for texture compatibility: Let $\{\mathbf{t}_i^0\}_{i=1}^{s_t}$ be a set of dense color samples on the template mesh of the deformable object expressed in barycentric coordinates with respect to vertices V_0 and projected to the reference image I_0. We optimize for the 3D vertex locations V_f so that the texture of each projected sample $\mathbf{t}_i^0$ at the reference frame matches the texture at its corresponding projected location $\mathbf{t}_i^f$ at frame f. As texture features $\phi(\cdot)$ we use the Gradient Based Descriptor Fields (GBDF) [5] that are robust under light changes.

$$E_T(V_f, V_0, \mathcal{M}_f) = \sum_{i=1}^{s_t} \left|\left|\phi(\mathbf{t}_i^0) - \phi(\mathbf{t}_i^f)\right|\right|_2^2 . \tag{11}$$

We consider only dense samples that given the last solution V_{f-1} for the deformable object project outside the hand mask $\mathcal{M}_f$ estimated in Section 3.1.

Preserve structure: The fifth term in Eq.(7) aims at preserving the template edge lengths, as those were defined in M_0. Thus, $E_S(V_f, \mathcal{E}_f, V_0)$ is defined as:

$$E_S(V_f, \mathcal{E}, V_0) = \sum_{(\mathbf{v}_i^f, \mathbf{v}_j^f) \in \mathcal{E}} \left(||\mathbf{v}_i^f - \mathbf{v}_j^f||_2 - ||\mathbf{v}_i^0 - \mathbf{v}_j^0||_2 \right)^2 . \tag{12}$$

Smoothness prior: To favor physically plausible deformations, especially in the occluded areas of the deformable object, we use a Laplacian-based regularizer as in [36,17]. We penalize non-rigid deformations away from the reference shape of the deformable object using the following error term where A is the Laplacian smoothing matrix defined based on the reference mesh M_0.

$$E_L(V_f, A) = ||AV_f||_2 . \tag{13}$$

Hand-object contact constraint: We assume that a hand may interact with/touch the deformable object by any of its fingertips represented by vertex indices $\mathbf{f} = \{f_i\}_{i=1}^{5}$. Thus, we define a contact configuration as a vector $Y = [y_1, y_2, y_3, y_4, y_5]$ where $y_i = 1$ if contact is assumed for the i-th fingertip and zero otherwise. For a hypothesized contact configuration Y, we want to minimize the distance of the fingertips assumed to be in contact from the surface of the deformable mesh. That leads to the following energy term

$$E_C(V_f, \theta_f, W_f, \mathbf{f}, Y) = \sum_{i=1}^{5} y_i ||\mathbf{w}_{f_i}^{f} - \mathbf{z}_i||_2, \tag{14}$$

where $\mathbf{z}_i$ is the closest point of the deformable mesh M_f to the i-th fingertip.

Optimization: At each frame f, the minimization problem of Eq.(6) is solved based on the Levenberg-Marquardt method as implemented in the Ceres solver [4] initialized with the inferred coordinates of the template vertices V_{f-1} at the previous frame $f - 1$ and the hand pose estimate θ'_f from Section 3.1. The weights quantifying the relative importance of the corresponding error terms were empirically set to $\lambda_J = 800$, $\lambda_{G\_h} = 10^3$, $\lambda_{G\_o} = 10^3$, $\lambda_F = 400$, $\lambda_T = 0.05$, $\lambda_S = 220 \times 10^3$, $\lambda_L = 0.3$ and $\lambda_C = 100$ for real data and were subsequently normalized by the number of subterms in the corresponding error term. The weights were kept constant throughout all quantitative experiments. The optimization runs for a maximum number of 100 iterations.

3.3 Optimal contact configuration selection

We minimize the energy function of Eq.(6) for all 32 possible contact configurations of the fingertips and end up with a solution M_f^i and $\theta_f^i, i = 1, \dots, 32$ for each contact configuration. We select the optimal contact configuration to be the one that results in the best fit of the object template mesh to the point cloud in the area around the hand. This is motivated by the fact that the correctness of a contact configuration can be judged by the consequences that it has on the estimation of the shape of the deformable object close to the contacting fingers. As an example, consider that a hand is at a certain distance from the object. The configuration $Y = [1, 1, 1, 1, 1]$ of full contact will result in "magnetizing" the deformable object towards the fingers and in a bad fit of the estimated object model to the point cloud. At the same time, the consequences of contacts are attenuated as we move away from the hand, especially for large surfaces.

More specifically, let $\{\mathbf{m}_f^{ij}\}_{j=1}^{N_{if}}$ be a set of 3D samples on the surface of M_f^i that project on the image I_f at least 10 and at most 30 pixels away from the contour of the hand rendered at pose θ_f^i. Let also $\{\mathbf{n}_f^{ij}\}_{j=1}^{N_{if}}$ be the closest points of $\{\mathbf{m}_f^{ij}\}_{j=1}^{N_{if}}$ on the point cloud P_f. The optimal contact configuration is the one that minimizes the Euclidean distance of the samples from the point cloud

$$i^* = \text{argmin}_i \sum_{j=1}^{N_{if}} ||\mathbf{m}_f^{ij} - \mathbf{n}_f^{ij}||_2. \tag{15}$$

Intuitively, that means that in the optimal contact configuration the fingers neither "magnetize"/attract nor penetrate the deformable object. The solution that we end up with at this point is $V''_f = M_f^{i^*}$ for the shape of the deformable object and $\theta''_f = \theta_f^{i^*}$ for the pose of the hand.

3.4 Solution refinement based on multiresolution texture features

We calculate the GBDF descriptors per pixel and we subsequently smooth out the descriptors using three Gaussian kernels with standard deviation $\sigma = 8, 4$ and 1 pixels respectively and kernel size 3σ, thus forming a feature pyramid. We minimize the energy function in Eq.(7) for each hypothesized contact configuration using the coarse descriptors ($\sigma = 8$). After we decide on the optimal contact configuration, we further refine the solution for the pose of the hand and the shape of the deformable object by minimizing Eq.(7) using the optimal contact configuration and the descriptors first for $\sigma = 4$ and then for $\sigma = 1$. Ideally, this step should have been part of the optimization loop. Practically, bringing it to the end of the optimization process reduces the computational requirements without significant degradation of tracking accuracy.

The resulting solutions V_f, θ_f will be used as initializations for fitting the hand and deformable objects to the data of the following frame $f + 1$.

4 Experimental Results

Evaluation datasets: So far, the Shape-from-Template community has treated human hands as occluders [17] and previous work on tracking hand-object interactions has focused mainly on objects with simple deformations. As a result, there are no compelling datasets containing complex deformable objects interacting with hands of varying articulation. We evaluate our method quantitatively using a set of synthetic sequences involving interaction between hands and deformable objects that we have generated with the Blender modeling software. We have also captured sequences using a Microsoft Kinect 2 [15] to show the applicability of our approach in real-world data[2].

Figure 2 shows sample frames from the synthetic sequences that we used for quantitative evaluation. We consider sequences where the hand pushes the object causing significant deformations (S1-S3) as well as sequences where the hand moves mainly along the surface of the object (S4-S6). We differentiate the sequences further by considering various scenarios such as rigid motion of the hand (S1), coarse hand articulation (powergrasp - S2), fine articulation (finger tapping - S3, S6), static hand and fine articulation (finger tapping - S4) and minimal variation in articulation (S5).

Evaluated methods: To showcase the importance of inferring the optimal contact configuration for tracking, we compare our "joint" optimization approach against two variations. In the first that we term "independent", tracking involves

[2] Available online at https://www.ics.forth.gr/cvrl/deformable_interaction/.

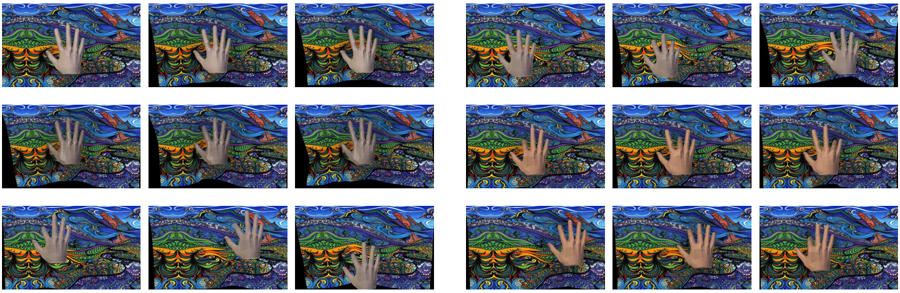


Fig. 2: Triplets of indicative frames for the synthetic sequences S1 to S6 (left to right, top to bottom). Each sequence consists of 23-51 frames and the template mesh consists of 529 vertices.

no contact constraints and corresponds to tracking the hand and deformable object independently, utilizing only the mask around the hand to determine the occluded area of the object. In the second that we term "fullcontact", tracking corresponds to assuming contact of all fingers with the deformable object.

Despite the fact that the work presented in [42] is the closest to ours in spirit, a comparison with it is not meaningful as it would require to represent deformable 3D meshes as tree-structured articulated objects. Instead, we compare with an implementation[3] of the RGBD method described in [41], by providing to this method the part of the deformable object that is occluded by the hand as an occlusion mask. We also provide results obtained from the state-of-the-art Shape-from-Template method by Ngo et al. [17] that fits the deformable object given the same occlusion mask. As mentioned in Section 2, this is an RGB-based method that does not consider depth information. Therefore, its results are not provided for direct comparison but rather for serving as a reference baseline.

Evaluation metrics: The quantitative evaluation of all methods in this study is performed using two error metrics. The first one, E_1, denotes the percentage of template vertices over all frames in a sequence whose inferred 3D locations are within distance T from their ground truth locations. We consider only the vertices $\mathbf{v}_i^f$, $i \in N_m^*$ that fall within the hand mask at each frame f. Because the methods in [17,41] do not track the human hand, in that case we use the mask coming from our joint tracking method. Thus,

$$E_1(T) = \frac{1}{|N_m^*| \cdot K} \sum_{i \in N_m^*} \sum_{f=1}^{K} g\left(||\mathbf{v}_i^f - \mathbf{x}_i^f||_2 < T \right), \tag{16}$$

where $g(x < T) = 1$ if $x < T$ and 0 otherwise. In Eq.(16), $\mathbf{x}_i^f$ is the ground truth location of vertex $\mathbf{v}_i^f$.

[3] Original implementation provided by the authors of [41] and modified to consider an occlusion mask.

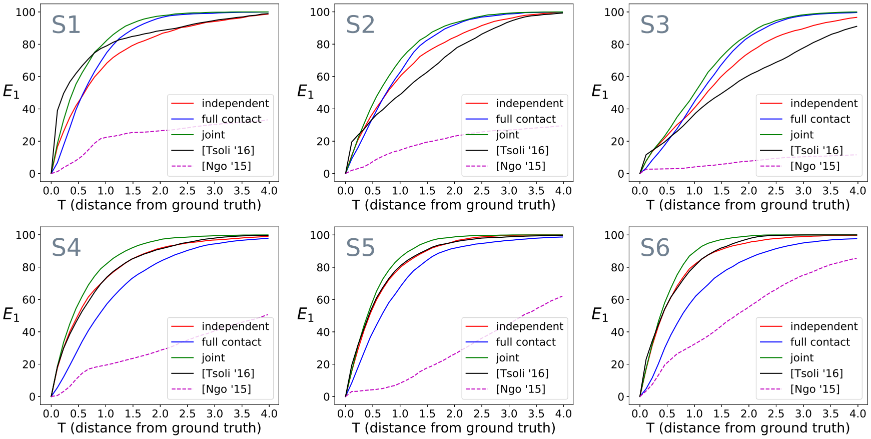


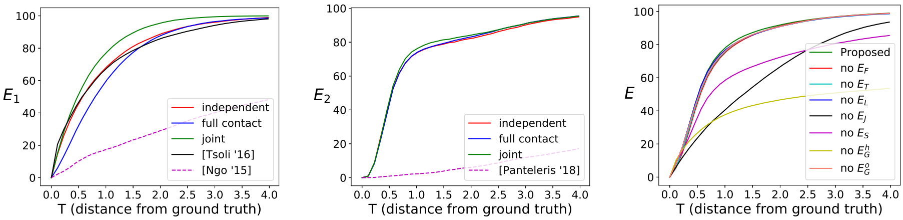
Fig. 3: Quantitative evaluation on synthetic data. For each sequence, we show the percentage of vertices inside the hand mask over all frames within Euclidean distance T from their ground truth location. Distance is expressed as a multiple of the length of a horizontal edge of the template. Each sequence consists of 23-51 frames, the template mesh consists of 529 vertices and the mask occupies on average around 20% of the surface of the deformable object.

In a similar way, we calculate the percentage of hand joints over all frames in a sequence whose estimated 3D locations $\mathbf{l}_i^f$ are within distance T from their ground truth 3D locations $\mathbf{h}_i^f$. Thus,

$$E_2(T) = \frac{1}{21 \cdot K} \sum_{i=1}^{21} \sum_{f=1}^{K} g\left(|||\mathbf{l}_i^f - \mathbf{h}_i^f||_2 < T \right). \tag{17}$$

Evaluation of object deformation tracking: Figure 3 shows the error metric E_1 for the synthetic sequences S1-S6. Distance T is expressed as a multiple of the length of a horizontal edge of the template. In S1-S6 this is equal to the width of the index finger of the hand model (T =2cm).

In sequences S1 to S3, where the hand interacts strongly with the object, we observe that tracking assuming full contact predicts more accurately the object's deformation in the occluded area than tracking the object and hand with no contact constraints (independent tracking). That holds true no matter how coarse or fine the articulation of the hand is. This is also the case that is most challenging for Ngo et al. [17]. Apart from the fact that the method proposed in [17] relies on RGB input and, thus, on less information about the observed geometry, the smoothness prior is not able to capture effectively the deformations of the object due to contact with the hand. The method in [41] is effective in the case of object deformations caused by a hand moving rigidly (S1), but underperforms when the articulation of the hand varies (S2-S3).



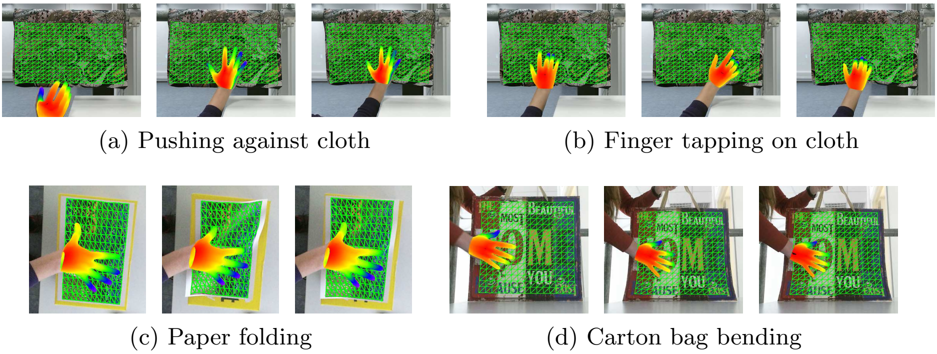
(a) Object (b) Hand joints (c) Ablative analysis

Fig. 4: (a,b) Aggregate results and (c) ablative analysis on synthetic sequences.

In sequences where the hand does not cause significant deformations of the object or contact points vary a lot in time due to the fine manipulation actions (S4-S6), making no contact assumptions is preferable than assuming full contact. The finer the articulation of the hand, the worst the full contact assumption performs (S5 vs S6). The method in [41] exhibits similar performance to tracking the deformable object with the variant of the proposed method that imposes no contact constraints. We also observe improved performance of the method in [17] relative to S1-S3. The global motion of the hand (static hand in S4 vs. hand moving along the object's surface in sequences S5, S6) does not influence the relative performance of the methods. However, joint tracking outperforms tracking assuming full contact, tracking with no contact and tracking based on [41] or based on [17]. The overall superior performance of our method is also highlighted in Figure 4a that shows the aggregate error over all sequences.

Tracking of the human hand: Figure 4b shows the E_2 metric for joint, independent and full contact tracking. Joint tracking exhibits very similar performance to the cases of independent and full contact tracking. This is attributed to the fact that in our synthetic sequences most hand joints are visible and, thus, there is strong evidence for the estimation of the 3D hand pose regardless of the variant used for joint hand-object tracking. The method in [24] constitutes the first step of our approach (Section 3.1). Given that it predicts the 3D locations of the joints solely from RGB input, its accuracy is low. As reported in [24], most of this error is along the camera optical axis which makes it possible to obtain an accurate occlusion mask, despite the 3D estimation error. Figure 4b shows that, when combined with depth and contact information in our joint optimization framework, the overall accuracy of 3D hand tracking is dramatically increased.

Ablative analysis: Figure 4c shows a combined error metric E denoting the percentage of vertices for both the object and the hand within distance T from their ground truth location for the objective function of Eq.(7) ('proposed') as well as when a certain term X is excluded ('no X'). Feature correspondences, texture and smoothness play a solution-refining role, while hand joints estimation, 3D structure-matching for the hand and mesh edges length preservation are critical. Note that some terms are correlated, i.e., E_F and E_T leverage color information and E_F and E_G^o take into account the observed geometry. Omitting



(a) Pushing against cloth

(b) Finger tapping on cloth

(c) Paper folding

(d) Carton bag bending

Fig. 5: Qualitative results on sequences obtained with MS Kinect 2.

only one term in these pairs has little impact on the overall performance, but omitting both of them will decrease tracking accuracy significantly. In any case, the best performance is obtained when all terms are employed.

Qualitative evaluation: Figure 5 shows indicative results on real data obtained with a Microsoft Kinect 2. Examples include motion of an open hand against a cloth, finger tapping on cloth, folding a paper with a single hand and bending a carton bag. The color coding on the hand denotes the relative depth of its vertices. The results are better viewed in the supplementary video[4].

5 Conclusions

Most of the deformable object tracking works either deal with such objects in isolation or consider a hand interacting with an object that can be effectively represented by a tree-structured articulated model. In this work, we presented the first method that tracks a complex deformable object represented as a 3D mesh, interacting with a hand. We formulated a joint optimization problem involving the minimization of an energy function whose terms depend on the appearance and the kinematics of the hand, the object and their interaction in the form of hand-object contact configurations. Thus, the hand is not treated as a distractor that occludes parts of the object, but as a source of valuable information. Evaluation on synthetic and real sequences illustrate the performance of the proposed method and show the accuracy gains over variants and other relevant solutions. Ongoing work aims at handling bimanual manipulation of deformable objects as well as contacts of the object at any point on the hand surface.

Acknowledgments

This work was partially supported by the EU project Co4Robots.

[4] Youtube video: https://youtu.be/JSOIy3D_5IO

References

1. Albrecht, I., Haber, J., Seidel, H.P.: Construction and Animation of Anatomically Based Human Hand Models. In: Eurographics symposium on Computer animation. p. 109. Eurographics Association (2003)
2. Ballan, L., Taneja, A., Gall, J., Van Gool, L., Pollefeys, M.: Motion capture of hands in action using discriminative salient points. In: European Conference on Computer Vision (ECCV). pp. 640–653. Springer (2012)
3. Bartoli, A., Gerard, Y., Chadebecq, F., Collins, T., Pizarro, D.: Shape-from-template. Pattern Analysis and Machine Intelligence, IEEE Transactions on **37**(10), 2099–2118 (2015)
4. Ceres Solver: http://ceres-solver.org/
5. Crivellaro, A., Lepetit, V.: Robust 3d tracking with descriptor fields. In: Conference on Computer Vision and Pattern Recognition (CVPR). No. EPFL-CONF-198219 (2014)
6. Garg, R., Roussos, A., Agapito, L.: A variational approach to video registration with subspace constraints. International journal of computer vision **104**(3), 286–314 (2013)
7. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3593–3601 (2016)
8. Hamer, H., Schindler, K., Koller-Meier, E., Van Gool, L.: Tracking a hand manipulating an object. In: IEEE International Conference On Computer Vision (ICCV). pp. 1475–1482. IEEE (2009)
9. Hilsmann, A., Eisert, P.: Tracking deformable surfaces with optical flow in the presence of self occlusion in monocular image sequences. In: Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on. pp. 6, 1 (2008), http://dx.doi.org/10.1109/CVPRW.2008.4563081
10. Kyriazis, N., Argyros, A.: Physically plausible 3d scene tracking: The single actor hypothesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9–16. IEEE (2013)
11. Kyriazis, N., Argyros, A.: Scalable 3d tracking of multiple interacting objects. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3430–3437. IEEE (2014)
12. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Quarterly of applied mathematics **2**(2), 164–168 (1944)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (Nov 2004)
14. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. Journal of the society for Industrial and Applied Mathematics **11**(2), 431–441 (1963)
15. Microsoft Kinect 2: https://developer.microsoft.com/en-us/windows/kinect
16. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In: Proceedings of International Conference on Computer Vision (ICCV). vol. 10 (2017)
17. Ngo, D.T., Park, S., Jorstad, A., Crivellaro, A., Yoo, C., Fua, P.: Dense image registration and deformable surface reconstruction in presence of occlusions and minimal texture. International Conference on Computer Vision (ICCV) (2015)

18. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3316–3324 (2015)
19. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient Model-based 3D Tracking of Hand Articulations using Kinect. In: BMVC. Dundee, UK (Aug 2011)
20. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: International Conference on Computer Vision (ICCV). pp. 2088–2095. IEEE (2011)
21. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Tracking the articulated motion of two strongly interacting hands. In: IEEE Computer Vision and Pattern Recognition (CVPR 2012). pp. 1862–1869. IEEE, Providence, Rhode Island, USA (June 2012)
22. Östlund, J.O.M., Varol, A., Ngo, T.D., Fua, P.: Laplacian Meshes for Monocular 3D Shape Recovery. In: European Conference on Computer Vision (2012)
23. Panteleris, P., Kyriazis, N., Argyros, A.A.: 3d tracking of human hands in interaction with unknown objects. In: British Machine Vision Conference (BMVC 2015). pp. 123–1. BMVA, Swansea, UK (September 2015)
24. Panteleris, P., Oikonomidis, I., Argyros, A.: Using a single rgb frame for real time 3d hand pose estimation in the wild (2018)
25. Parashar, S., Pizarro, D., Bartoli, A., Collins, T.: As-rigid-as-possible volumetric shape-from-template. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 891–899 (2015)
26. Petit, A., Lippiello, V., Siciliano, B.: Tracking an elastic object with an rgb-d sensor for a pizza chef robot
27. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1106–1113 (2014)
28. Romero, J., Kjellstrom, H., Kragic, D.: Monocular real-time 3d articulated hand pose estimation. IEEE-RAS Int'l Conf. on Humanoid Robots (Dec 2009). https://doi.org/10.1109/ICHR.2009.5379596, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5379596
29. Salzmann, M., Lepetit, V., Fua, P.: Deformable surface tracking ambiguities. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. pp. 1–8. IEEE (2007)
30. Schulman, J., Lee, A., Ho, J., Abbeel, P.: Tracking deformable objects with point clouds. In: Proceedings of the International Conference on Robotics and Automation (ICRA) (2013)
31. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., et al.: Accurate, robust, and flexible real-time hand tracking. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 3633–3642. ACM (2015)
32. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2 (2017)
33. Sinha, A., Choi, C., Ramani, K.: Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4150–4158 (2016)
34. Sridhar, S., Mueller, F., Zollhöfer, M., Casas, D., Oulasvirta, A., Theobalt, C.: Real-time joint tracking of a hand manipulating an object from rgb-d input. In: European Conference on Computer Vision. pp. 294–310. Springer (2016)

35. Sridhar, S., Oulasvirta, A., Theobalt, C.: Interactive markerless articulated hand motion tracking using rgb and depth data. In: IEEE International Conference on Computer Vision (ICCV). pp. 2456–2463. IEEE (2013)
36. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. In: ACM Transactions on Graphics (TOG). vol. 23, pp. 399–405. ACM (2004)
37. Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., Pauly, M.: Robust articulated-icp for real-time hand tracking. In: Computer Graphics Forum. vol. 34, pp. 101–114. Wiley Online Library (2015)
38. Tang, D., Jin Chang, H., Tejani, A., Kim, T.K.: Latent regression forest: Structured estimation of 3d articulated hand posture. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3786–3793 (2014)
39. The Blender open source 3D creation suite: https://docs.blender.org/
40. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Transactions on Graphics (ToG) **33**(5), 169 (2014)
41. Tsoli, A., Argyros, A.: Tracking deformable surfaces that undergo topological changes using an rgb-d camera. In: Proceedings of International Conference on 3D Vision (3DV). Stanford University, CA, USA (October 2016)
42. Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J.: Capturing hands in action using discriminative salient points and physics simulation. International Journal of Computer Vision **118**(2), 172–193 (2016)
43. Tzionas, D., Gall, J.: 3d object reconstruction from hand-object interactions. In: International Conference on Computer Vision (ICCV). pp. 729–737 (Dec 2015)
44. Tzionas, D., Srikantha, A., Aponte, P., Gall, J.: Capturing hand motion with an rgb-d sensor, fusing a generative model with salient points. In: German Conference on Pattern Recognition. pp. 277–289. Springer (2014)
45. Wan, C., Probst, T., Van Gool, L., Yao, A.: Crossing nets: Combining GANs and VAEs with a shared latent space for hand pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
46. Wan, C., Yao, A., Van Gool, L.: Hand pose estimation from local surface normals. In: European Conference on Computer Vision. pp. 554–569. Springer (2016)
47. Wuhrer, S., Lang, J., Shu, C.: Tracking complete deformable objects with finite elements. In: 3DIMPVT. pp. 1–8. IEEE Computer Society (2012), http://dblp.uni-trier.de/db/conf/3dim/3dimpvt2012.html#WuhrerLS12
48. Ye, Q., Yuan, S., Kim, T.K.: Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In: European Conference on Computer Vision. pp. 346–361. Springer (2016)