InteractionFusion: Real-time Reconstruction of Hand Poses and Deformable Objects in Hand-object Interactions

HAO ZHANG, BNRist and School of Software, Tsinghua University, China
ZI-HAO BO, BNRist and School of Software, Tsinghua University, China
JUN-HAI YONG, BNRist and School of Software, Tsinghua University, China
FENG XU, BNRist and School of Software, Tsinghua University, China

Hand-object interaction is challenging to reconstruct but important for many applications like HCI, robotics and so on. Previous works focus on either the hand or the object while we jointly track the hand poses, fuse the 3D object model and reconstruct its rigid and nonrigid motions, and perform all these tasks in real time. To achieve this, we first use a DNN to segment the hand and object in the two input depth streams and predict the current hand pose based on the previous poses by a pre-trained LSTM network. With this information, a unified optimization framework is proposed to jointly track the hand poses and object motions. The optimization integrates the segmented depth maps, the predicted motion, a spatial-temporal varying rigidity regularizer and a real-time contact constraint. A nonrigid fusion technique is further involved to reconstruct the object model. Experiments demonstrate that our method can solve the ambiguity caused by heavy occlusions between hand and object, and generate accurate results for various objects and interacting motions.

CCS Concepts: • **Computing methodologies → Shape modeling**.

Additional Key Words and Phrases: hand tracking, hand-object interaction, non-rigid motion, model reconstruction

ACM Reference Format:
Hao Zhang, Zi-Hao Bo, Jun-Hai Yong, and Feng Xu. 2019. InteractionFusion: Real-time Reconstruction of Hand Poses and Deformable Objects in Hand-object Interactions. *ACM Trans. Graph.* 38, 4, Article 48 (July 2019), 11 pages. https://doi.org/10.1145/3306346.3322998

1 INTRODUCTION

Reconstructing 3D motions of human hands is an important topic in computer vision and graphics due to its numerous applications in Human-Computer Interaction (HCI), robotics, rehabilitation, behavior analysis, virtual and augmented reality, and so on. There are many works focusing on the tracking of isolated hands, which is useful for a few applications like gesture recognition and control. However, human hands are majorly used for interacting with the environment or manipulating objects. Comparing with isolated hands, the reconstruction of interacting motions is more eagerly required.

Authors' addresses: Hao Zhang, BNRist and School of Software, Tsinghua University, China, zhanghao16@mails.tsinghua.edu.cn; Zi-Hao Bo, BNRist and School of Software, Tsinghua University, China; Jun-Hai Yong, BNRist and School of Software, Tsinghua University, China; Feng Xu, BNRist and School of Software, Tsinghua University, 30 Shuangqing Rd, Haidian Qu, Beijing Shi, China, feng-xu@tsinghua.edu.cn.

0730-0301/2019/7-ART48 $15.00
https://doi.org/10.1145/3306346.3322998

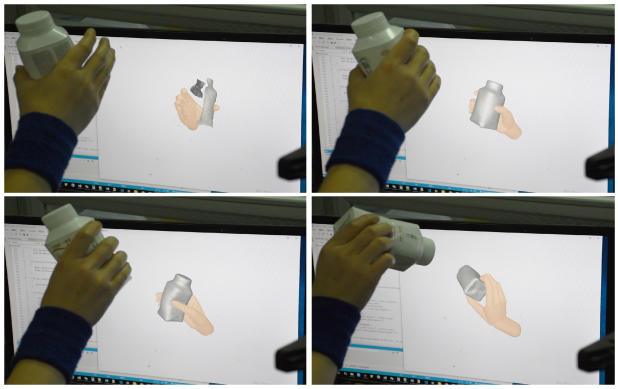

Fig. 1. Real-time reconstruction of hand-object interactions. The top two figures indicate the fusion of the object geometry from incomplete to complete. The bottom two figures show the nonrigid deformation and rigid motion of the object caused by interactions. More results are shown in the result section and the accompanying video.

In literature, besides pure hand tracking [Han et al. 2018; Taylor et al. 2016; Tkach et al. 2017], there are many works focusing on hand-object interactions. Some works assume known object shape and majorly focus on the interacting motions of hands [Ballan et al. 2012; Hamer et al. 2009; Kyriazis and Argyros 2013, 2014; Oikonomidis et al. 2011; Sridhar et al. 2016]. These techniques need to pre-model the objects and do not allow shape changes of objects. Recently, some works can handle objects with articulated motions [Tzionas et al. 2016] or even nonrigid motions [Petit et al. 2018; Tsoli and Argyros 2018]. To solve the object motions, these techniques require offline processing and initial templates of the objects, which limits their applications in reality. Techniques for in hand reconstruction aim to reconstruct the shape of in hand objects and thus do not require templates, but they only handle static shapes, not motions of objects. In general, there is no existing work that simultaneously reconstructs the 3D motion of human hands and the shapes of the manipulated objects as well as their motions (rigid or nonrigid), not to mention achieving all of them in real time.

There are some key challenges for the aforementioned full reconstruction of hand-object interactions. Hand tracking itself is difficult due to the ambiguity caused by complex motions, lack of geometry/texture features and self-occlusion. When hand interacts with objects, the ambiguity further increases due to the much heavier occlusions. The same situation happens to the objects, which may always be occluded by fingers and the palm. Furthermore, for objects, we need to not only solve their rigid and nonrigid motions,

but also reconstruct their static geometry models. Otherwise, we still need to pre-scan the geometry models for the following motion estimation. Furthermore, the geometries and motions of objects, and the articulated hand poses form a high dimensional solution space, which requires adequate and efficient regularizers to solve the ambiguity and powerful optimization algorithms to accomplish the solving in real time.

In this paper, we propose a novel technique to solve the mentioned ambiguities and achieve the full reconstruction of hand-object interactions in real time. First, we build a recording system with two depth cameras that record interacting motions from opposite directions. In this manner, we do not rely on a very heavy capture system but still record as much information as possible. Then we combine neural networks and generative tracking models to solve the reconstruction problem. We first train a Deep Neural Network (DNN) to segment hand and object from the depth input, which will help to avoid the mutual interference between them in the following tracking step. Then we train a Long Short Term Memory (LSTM) network to predict the hand pose in one frame from its previous frames. This is also important because for some interacting poses, fingers may be totally occluded for both cameras, and this network considers motion priors to predict a reasonable pose and thus help alleviate the ambiguity. In general, the neural networks extract coarse information as much as possible by learning from data. Then the generative model further solves the ambiguity to get the final accurate reconstruction by fusing two-frame information and involving delicately designed regularizers. Besides the aforementioned pose prediction, we propose a rigidity regularizer with spatial-temporal varying weights, which allows non-smooth deformations around contact regions and also guarantees smooth deformations on the other regions. This regularization helps to generate local nonrigid motions and contributes to solving the loop closure problem in fusing the geometry model. We also propose a contact regularizer in a new formulation to generate more accurate and physically plausible results. The new formulation helps to keep real-time performance. The contributions of our technique can be summarized as follows:

- To the best of our knowledge, this is the first system that simultaneously achieves hand tracking, object fusion and nonrigid motion tracking for hand-object interactions. Furthermore, it achieves real-time performance.
- We propose an LSTM-based predictor to predict hand poses in a sequence and a novel interaction term to estimate more accurate and realistic interactive motions.
- We propose a generative model to solve the reconstruction problem by fusing the point cloud of two frames and involving pose, object and joint pose-and-object regularizers in a unified optimization framework.

2 RELATED WORKS

In this section, we will majorly discuss papers focusing on hand-object interactions, which is also the topic of this paper. A comprehensive discussion for isolated hand tracking and two hand tracking is not included here and can be found in [Panteleris and Argyros 2017].

2.1 Hand Tracking with Interactions

As interactions cause a lot of occlusions to both hand and objects, some works use multiple cameras to record multi-view information [Ballan et al. 2012; Oikonomidis et al. 2011; Wang et al. 2013]. Later, with the development of depth sensing, a single RGB-D sensor is used to perform hand tracking with interactions [Kyriazis and Argyros 2013, 2014]. However, these works still need to know the shape of the manipulated objects and require offline processing, which restricts their usage in many applications. Data-driven techniques are also proposed for this task, such as [Romero et al. 2010], which uses database searching to find output hand poses. As the size of the database is limited, complex hand-object interactions cannot be correctly reconstructed. Segmentation is also useful for hand pose estimation with interactions. As pure hand regions can be identified, poses can be estimated by SVM classification [Rogez et al. 2015a,b] and hand part classification [Hamer et al. 2009; Sridhar et al. 2016]. Recently, recurrent neural network (RNN) is used to recover failed detection caused by occlusions of the hand itself [Quach et al. 2016]. The RNN computes joint coordinate directly and contributes to handling occlusion situations. Choi et al. [2017] use a convolutional neural network (CNN) to localize hand and unknown objects, and reconstruct hand poses by identifying the global orientation of objects and grasp type of hands. Mueller et al. [2017] also explore CNN in hand localization and joint detection, but majorly focus on egocentric RGB-D input. Panteleris and Argyros [2017] use short-baseline stereo cameras to reconstruct hands when interacting with objects or another hand, by exploring color consistency. Taylor et al. [2017] propose an articulated signed distance function to avoid explicit correspondences and achieve extremely fast hand pose estimation. Mueller et al. [2018] use GAN to produce annotated RGB images and achieve a real-time hand tracking system from unconstrained monocular RGB images. In general, techniques for pose estimation during hand-objects interactions have been highly developed in the literature, but they majorly focus on hand but not objects. The objects are assumed to be with known shapes or not reconstructed in the final results.

2.2 In Hand Reconstruction

On the contrary, some other works majorly focus on the objects in interactions. A variety of methods [Krainin et al. 2011; Rusinkiewicz et al. 2002] use Iterative Closest Point (ICP) to register the scanned point cloud of multiple frames to reconstruct the final object shape. Weise et al. [2008] combine texture and geometry information together to pursuit better reconstruction. Weise et al. [2011] propose an online loop closure method to remove the accumulated errors and reconstruct the final complete 3D model. Yuheng Ren et al. [2013] propose a probabilistic framework to jointly handle tracking and reconstruction problems for objects in hand. However, these methods just use hands as a turntable and discard them in the reconstruction, and thus they are not able to handle objects with neither geometry features nor textures. Recently, some works consider hand motion in the reconstruction. Tzionas and Gall [2015] extract hand motions and use contact information to predict object motions to solve the problem of lacking geometry or texture features, but this work still only considers rigid objects. Recently, Petit et al. [2018] reconstruct

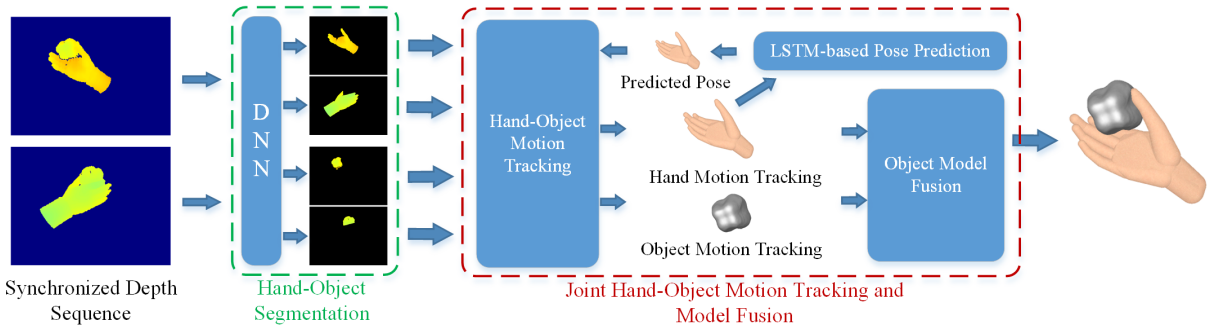

Fig. 2. Overview of our tracking pipeline.

the in-hand nonrigid motions of an object rather than the static shape of the object. However, they require to scan the initial shape.

2.3 Joint Hand-object Reconstruction

Different from all the previous methods, this paper focuses on simultaneously reconstructing 3D motion of the hand and the interacted object, including its shape and its nonrigid motions in the sequence. Panteleris et al. [2015] assume a rigid object, and reconstruct its geometry as well as 3D hand motions. A variety of methods [Schmidt et al. 2015; Tzionas et al. 2016] reconstruct articulated motions of objects with hand poses, but they require an initial shape of the object, and thus always require a 3D scan of a new object. Tsoli and Argyros [2018] make a step further to handle nonrigid motions by exploring appearance features, but again, their method requires an initial shape of the object. In this paper, we get rid of all these constraints to reconstruct the shape and nonrigid motions of an object, as well as the motions of the hand manipulating the object. Furthermore, we achieve real-time performance.

3 OVERVIEW

Our system uses two depth cameras to record interacting motions from two opposite directions. The two cameras are pre-calibrated and the recorded depth streams are synchronized to form a sequence of paired depth frames. To keep real-time performance, the system runs in a frame-by-frame manner, i.e., processing depth pairs one after another. Fig.2 illustrates the processing of one depth pair. The first step is the Hand-Object Segmentation, in which the hand and the object are segmented apart from each other through a trained network. Notice that the two frames in the pair are segmented in parallel as the network is trained to segment one frame independently. Meanwhile, The reconstructed hand poses of the previous depth pairs are fed into another trained LSTM model to predict a hand pose of this pair.

Following are the Joint Hand-Object Motion Tracking and Object Model Fusion. Firstly, the motions of the hand and the object are estimated by solving a unified optimization problem to fit the segmented depth, the predicted pose prior and other proposed regularization terms. After the motion estimation, we fuse the masked depth of the object into its current geometry model, leading to a more smooth and complete reconstruction. Now, the hand and object of this frame are both obtained and can be used in the processing of the next frame.

4 PRELIMINARY

In this section, we present the notations and mathematical representations used in this paper. The synchronized paired depth sequence is denoted as $\{\mathcal{D}_0^t, \mathcal{D}_1^t\}$ (t indicates the frame label). Here 0 and 1 indicate the left and right depth camera respectively. It should be noted that consumer depth sensors could record a color sequence along with the depth sequence, but in this work, we only use color information to detect the bracer and further to get the hand-object depth by the method introduced in [Tkach et al. 2016]. The reconstructed object is represented as a fused static model $\mathcal{S}$ and its per-frame non-rigid motion is represented as a motion field $\mathcal{W}$. The per-frame hand motion is represented as joint rotations with totally 28 Degrees of Freedom (DOF) $\boldsymbol{\theta}$ (including 6 DOF for global rotations and translations).

To be more specific, following the work [Newcombe et al. 2015], we use Truncated Signed Distance Function (TSDF) to represent the static model of object $\mathcal{S}$, which is defined as $\mathcal{S} = \{d(\mathbf{x}), w(\mathbf{x})\}$, where $\mathbf{x}$ is a coordinate in a canonical frame, $d(\mathbf{x})$ indicates the signed distance from $\mathbf{x}$ to its closest surface point and $w(\mathbf{x})$ represents the confidence of $d(\mathbf{x})$. We use the coordinate system of the left depth camera as the world coordinate system and its first recorded frame as the canonical frame to represent $\mathcal{S}$. The surface mesh $\mathcal{M}$ of the object can be extracted by the zero-valued surface of $\mathcal{S}$ as:

$$\mathcal{M} = \{(\mathbf{v}, \mathbf{n}) | d(\mathbf{v}) = 0, \mathbf{n} = \frac{\nabla d(\mathbf{v})}{\| \nabla d(\mathbf{v}) \|_2}\} \tag{1}$$

Since $\mathcal{S}$ is represented in the canonical frame, $\mathcal{M}$ is called the canonical model. Given a motion field $\mathcal{W}(\mathbf{x})$, a live model $\mathcal{M}_l$ is represented as:

$$\mathcal{M}_l = \{(\mathbf{v}_l, \mathbf{n}_l) | \mathbf{v}_l = \mathcal{W}(\mathbf{v})\mathbf{v}, \mathbf{n}_l = \mathcal{W}(\mathbf{v})\mathbf{n}\} \tag{2}$$

The live model $\mathcal{M}_l$ is a deformed shape of the object which is used to represent the geometry of the object in a live recorded frame.

To represent the non-rigid deformation of object $\mathcal{W}(\mathbf{x})$, we follow the method in DynamicFusion [Newcombe et al. 2015]. Specifically, a node-based motion representation method is used to represent $\mathcal{W}(\mathbf{x})$. At first, a set of nodes are sampled uniformly on the surface

of the canonical model $\mathcal{M}$, and we define a compact motion field on the nodes: $\mathcal{W}_\mathcal{N} = \{[\mathbf{p}_i, \mathbf{T}_i]\}$. $\mathbf{p}_i \in \mathbb{R}^3$ is the coordinate of the ith node in the canonical frame. $\mathbf{T}_i \in \mathbf{SE}(3)$ is the transformation of the ith node. Then, the motion field $\mathcal{W}(\mathbf{x})$ can be interpolated by the node motions:

$$\mathcal{W}(\mathbf{x}) = \mathbf{SE}3(\sum_{k \in N(\mathbf{x})} \omega_k \mathbf{dq}_k) \tag{3}$$

where $N(\mathbf{x})$ contains the indexes of nodes which have influence on $\mathbf{x}$, $\mathbf{dq}_k$ is the dual quaternion of $\mathbf{T}_k$. $\omega_k = exp(-\|\mathbf{x} - \mathbf{p}_k\|^2/(2\sigma^2))$ decides how much the kth node has influence on $\mathbf{x}$. $\mathbf{SE}3(\cdot)$ converts a dual-quaternion back to a transformation matrix.

We follow [Tkach et al. 2016] to use sphere-mesh to model and track a hand. A sphere mesh is the mesh surface of the convex-hull of some spheres, and there are two types of basic sphere meshes used to represent a hand. One type (type 1) is constructed by two spheres, and the other (type 2) is constructed by three spheres. The geometry of a hand is modeled by assembling 30 basic sphere meshes, including 15 type 1 sphere meshes to majorly model fingers and 15 type 2 sphere meshes to model palm. Different poses are represented as different relative motions between the neighboring basic sphere meshes. Therefore, the motion of a hand is also defined on the sphere meshes. After pre-calibrating the geometry of a hand (deciding the relative position of the spheres and their sizes), the motion of a hand is represented as rotations at hand joints $\boldsymbol{\theta}$. Please refer to [Tkach et al. 2016] to get more details.

5 METHOD

In this section, we will introduce the key steps in our method. For a new time instance t, we have the input pair $\{\mathcal{D}_0^t, \mathcal{D}_1^t\}$, the current static model $\mathcal{S}$ as well as the motion field $\mathcal{W}^{t*}$ and the hand motion $\boldsymbol{\theta}^{t*}$ of all the previous frame $t*$ ($t* = 0, ...t-1$). The whole method in this section will generate our final output including $\mathcal{W}^t$, $\boldsymbol{\theta}^t$ and an updated $\mathcal{S}$.

5.1 DNN based Hand-Object Segmentation

In this step, we segment an input depth map into three parts: hand, object and background. This step is important because otherwise we may easily use the depth of the object to estimate the motion of the hand, and vice versa, as the hand and the object are closely interacted. To perform the segmentation, we use a specialized deep learning network - DenseAttentionSeg [Bo et al. 2019]. The network consists of an encoder and a decoder, together with an attention mechanism to enhance the skip-connection between them. We shrink the network image size to 320x240 and use a multi-thread pipeline to increase the processing speed.

5.2 LSTM-based Pose Prediction

Although we use two cameras to avoid occlusion, there are still situations that both the cameras cannot catch enough data for some fingers. Besides, the segmentation model may also have defects in some frames, which may cause jittering and wrong pose estimation in the tracking step. So we use an LSTM-based network [Hochreiter and Schmidhuber 1997] to learn a motion prior of hand with interacting motions, which is considered as a corrective model for the Joint Hand-Object Motion Tracking system. The prior helps to eliminate high-frequency pose jittering and predict reasonable hand poses where there are occlusions or segmentation errors. The network consumes the hand poses of previous frames solved by the Joint Hand-Object Motion Tracking system, then outputs a predicted hand pose based on the previous observations and learned strategy, which will be used by the tracking system in the following frames.

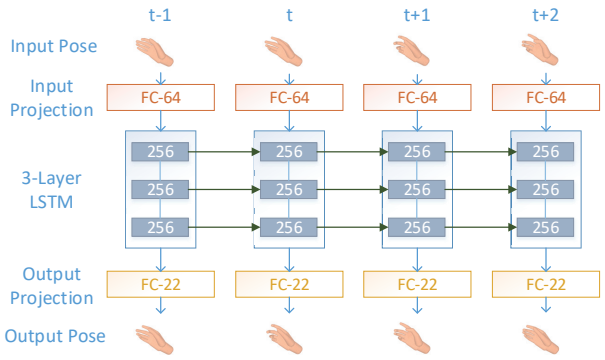

Fig. 3. LSTM pose prediction network. The input and output dimensions of the hand predictor are both 22. We use a stacked 3-layer LSTM, each of which has 256 units. FC means fully connected layer followed by the number of the output dimension.

Our LSTM network takes 22 DoFs as input and predicts a new 22 DoFs as output. Although there are 28 DoFs in [Tkach et al. 2016], we don't consider the translation and rotation degrees of the palm (6 DoFs in total) because it barely contains noise. The values of all DoFs are between $-\pi$ and π. Using this pose representation is more robust and reasonable than using joint coordinates directly [Quach et al. 2016]. The network structure is shown in Fig 3, including an input projection layer to encode hand poses, a 3-layer LSTM of dimension 256 to handle the pose sequence and an output projection layer to decode the feature back into a pose vector.

We construct the training dataset of the LSTM as follows. First, in order to make the network learn a motion space of the tracked hands in interactions, we collect many hand pose sequences with different interactive motions, including manipulating real objects (tracked by our method but without LSTM), and pretending to manipulate objects (tracked by the baseline [Tkach et al. 2016]). We manually check the tracked poses by comparing the point cloud and the tracked hand, and select the sequences which are clear and accurate. Then, to model the possible errors generated by the tracking system, we add random gaussian noises to the sequences and build the input of our network. To be specific, we randomly select no more than 3 DoFs in each frame. Then we add to the i-th DoF with a zero-mean gaussian noise whose standard deviation σ is:

$$\sigma_i = \alpha \cdot c_i \tag{4}$$

where c_i is used to control the magnitude of the added error of the i-th DoF. c_i is uniformly sampled between 0.41-0.91 if the Dof is selected, otherwise, c_i is sampled between 0.01-0.11. α is set to 0.7 which means the mean deviation of noises for the selected Dof is about 0.45. Finally, we smooth the originally tracked pose sequences

to build the output ground truth of our LSTM, in consideration of time continuity and possible high-frequency jittering noises when tracking. The window size of the linear weighted mean smoothing filter is 7 frames.The predicted hand pose at frame t is denoted as $\boldsymbol{\theta}_p^t$.

5.3 Joint Hand-Object Motion Tracking

As previously mentioned, our system operates in a tracking procedure. Therefore, for frame t, we have the current static model S in the canonical frame, the motion field of the object $\mathcal{W}_{\mathcal{N}}^{t-1}$ and the hand pose $\boldsymbol{\theta}^{t-1}$ of the previous frame. Using the newly recorded depth in frame t and the segmentation and pose prediction results, the algorithm presented in this section will reconstruct the motion of the object $\mathcal{W}_{\mathcal{N}}^t$ and the hand pose $\boldsymbol{\theta}^t$ at this frame. We propose a unified optimization framework for best fitting the input depth as well as satisfying all our proposed regularization terms. The energy function is formulated as follows:

$$E_{\text{tol}}(\mathcal{W}_{\mathcal{N}}^t, \boldsymbol{\theta}^t) = E_{\text{obt}}(\mathcal{W}_{\mathcal{N}}^t) + E_{\text{hand}}(\boldsymbol{\theta}^t) + \omega_{\text{itc}} E_{\text{itc}}(\mathcal{W}_{\mathcal{N}}^t, \boldsymbol{\theta}^t) \quad (5)$$

where $E_{\text{obt}}(\mathcal{W}_{\mathcal{N}}^t)$ is the energy term defined on the object, $E_{\text{hand}}(\boldsymbol{\theta}^t)$ is the term for the hand, and $E_{\text{itc}}(\mathcal{W}_{\mathcal{N}}^t, \boldsymbol{\theta}^t)$ is the interaction term related to both the hand and the object. Prior to solving the energy function, we first extract a surface mesh $\mathcal{M}$ from the current static model $\mathcal{S}$, and the node graph $\mathcal{G}$ is obtained via graph generation or graph updating (Details can be found in [Newcombe et al. 2015]).

5.3.1 Energy Term of the Object. The energy term, which is purely related to the deformation of the object, is formulated as:

$$E_{\text{obt}}(\mathcal{W}_{\mathcal{N}}^t) = E_{\text{o-dep}}(\mathcal{W}_{\mathcal{N}}^t) + E_{\text{o-silh}}(\mathcal{W}_{\mathcal{N}}^t) + E_{\text{o-reg}}(\mathcal{W}_{\mathcal{N}}^t) \quad (6)$$

where $E_{\text{o-dep}}$ and $E_{\text{o-silh}}$ are data terms that constrain the motion of object to be consistent with the depth input, $E_{\text{o-reg}}$ regularizes the resolved motion to be as locally rigid as possible.

The term $E_{\text{o-dep}}$ is to constrain that the reconstructed object should be consistent with the recorded depth of the object, which is formulated by a point-to-plane energy:

$$E_{\text{o-dep}}(\mathcal{W}_{\mathcal{N}}^t) = \sum_{s=0}^{1} \sum_{(\mathbf{v}_l, \mathbf{u}^t) \in C_{\text{dep}}^s} (\mathbf{n}_{\mathbf{u}^t}^{\mathrm{T}} (\mathbf{v}_l - \mathbf{u}^t))^2 \quad (7)$$

where s is the index of a sensor, $\mathbf{u}^t$ and $\mathbf{n}_{\mathbf{u}^t}$ are the coordinate and normal of a 3D point captured at frame t in the global coordinate system. C_{dep}^s indicates the correspondences between the vertices of the live model $\mathcal{M}_l$ and the point cloud captured by sensor s, which is built by the projective ICP method [Rusinkiewicz and Levoy 2001]. Notice that a minimum distance threshold for $(\mathbf{u}^t, \mathbf{v}_l)$ and a minimum cosine threshold for $(\mathbf{n}_{\mathbf{u}^t}, \mathbf{n}_l)$ are used to eliminate the outlier correspondences as [Newcombe et al. 2015] did.

$E_{\text{o-silh}}(\mathcal{W}_{\mathcal{N}}^t)$ is to penalize the model from lying outside the boundary of the object in the 2D image domain. The energy is constructed by 2D point-to-point constraint:

$$E_{\text{o-silh}}(\mathcal{W}_{\mathcal{N}}^t) = \sum_{s=0}^{1} \sum_{(\mathbf{v}_l^{\mathbf{o}}, \mathbf{u}_{\text{e}}^t) \in C_{\text{silh}}^s} \|\Pi_s \mathbf{v}_l^{\mathbf{o}} - \Pi_s \mathbf{u}_{\text{e}}^t\|_2^2 \quad (8)$$

where $\mathbf{v}_l^{\mathbf{o}}$ is a vertex of the live model whose 2D projection on sensor s is outside the mask of the object, $\mathbf{u}_{\text{e}}^t$ is a recorded depth point projected onto the edge of the mask. Π_s is the 3D to 2D projection for the sensor s. Here, we use orthogonal projection to maintain a linear problem. $\mathbf{v}_l^{\mathbf{o}}$ and $\mathbf{u}_{\text{e}}^t$ form a correspondence pair in C_{silh}^s in the condition that $\Pi_s \mathbf{u}_{\text{e}}^t$ has the minimum 2D distance to $\Pi_s \mathbf{v}_l^{\mathbf{o}}$ comparing with other depth points.

$E_{\text{o-reg}}(\mathcal{W}_{\mathcal{N}}^t)$ is to regularize the deformation of the object. Since we assume the object could have some nonrigid motions, strong smooth or rigid regularization is not adequate. However, we observe that for an object interacted by hands, non-rigidity does not happen everywhere. It majorly happens around contact points where external forces may be added to the object. For other regions which are far from contact points, there will be no external forces except the gravity. For these regions, they will majorly follow rigid motions even though the object is a nonrigid one. Based on this observation, for each node in the motion graph of the object, we calculate its distance to its nearest contact point and decide the regularization based on this distance. If a node is far from any contact points, it will have a strong rigid regularization, otherwise, it will have a weak regularization. In this manner, possible nonrigid motions can be correctly generated while rigid regions still have enough constraints to be robustly estimated with occlusions. Furthermore, the regularization contributes to reduce error accumulation and thus is helpful for achieving loop closure. The energy term is designed as follows:

$$E_{\text{o-reg}}(\mathcal{W}_{\mathcal{N}}^t) = \sum_{j \in \mathcal{G}} \sum_{i \in \mathcal{N}_j} \frac{\Omega(\mathbf{p}_i) + \Omega(\mathbf{p}_j)}{2} \|\mathbf{T}_j^t \mathbf{p}_j - \mathbf{T}_i^t \mathbf{p}_j\|_2^2 \quad (9)$$

where $\mathcal{N}_j$ contains the adjacent nodes of node j in the node graph $\mathcal{G}$. $\Omega(\mathbf{p})$ is the smooth coefficient determined by the distance of node $\mathbf{p}$ to its nearest contact point on the surface of the object. The method to detect contact points is formulated in Sec. 5.3.3.

5.3.2 Energy Term for Hand Tracking. $E_{\text{hand}}(\boldsymbol{\theta}^t)$ is the energy term which is only related to hand poses and is formulated as:

$$\begin{aligned} E_{\text{hand}}(\boldsymbol{\theta}^t) =& \omega_{\text{d2m}} E_{\text{d2m}}(\boldsymbol{\theta}^t) + \omega_{\text{m2d}} E_{\text{m2d}}(\boldsymbol{\theta}^t) + \omega_{\text{pose}} E_{\text{pose}}(\boldsymbol{\theta}^t) \\ &+ \omega_{\text{lim}} E_{\text{lim}}(\boldsymbol{\theta}^t) + \omega_{\text{colli}} E_{\text{colli}}(\boldsymbol{\theta}^t) \\ &+ \omega_{\text{temp}} E_{\text{temp}}(\boldsymbol{\theta}^t, \boldsymbol{\theta}^{t-1}) + \omega_{\text{lstm}} E_{\text{lstm}}(\boldsymbol{\theta}^t) \end{aligned} \quad (10)$$

where $E_{\text{d2m}}(\boldsymbol{\theta}^t)$ measures the distance between the recorded depth points and surface points on hand. $E_{\text{m2d}}(\boldsymbol{\theta}^t)$ is to prevent the tracked hand from lying outside of the visual hull of the real hand. $E_{\text{pose}}(\boldsymbol{\theta}^t)$ and $E_{\text{lim}}(\boldsymbol{\theta}^t)$ is to involve the pose prior and ensure the joint limits are not violated. E_{colli} is to prevent collisions between fingers, and E_{temp} is to maintain the time continuity of hand poses. For more details of these terms, please refer to [Tkach et al. 2016].

The novel part in E_{hand} is the prediction term $E_{\text{lstm}}(\boldsymbol{\theta}^t)$. As mentioned in the introduction, for hand object interactions, occlusions

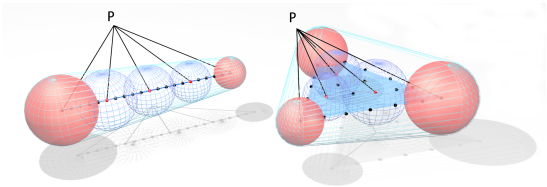

Fig. 4. Intersection detection for sphere meshes. The red spheres are the original spheres and the blue wireframe spheres are some new spheres. We have not drawn all new spheres but only their centers as dots on the "skeleton" of sphere meshes. For a point P, we calculate their distance to all the centers to construct the interaction term. The left sphere mesh is of type 1 and the right type 2.

become much stronger. It is not enough to solve the caused ambiguity by the traditional regularization terms like E_{pose}, E_{lim}, E_{colli} and E_{temp}. Some hand joints may be occluded leading to wrong poses which are still in the possible pose space (E_{pose} cannot help) and do not exceed the joints limits or collide to other fingers (E_{lim} and E_{pose} cannot help). And furthermore, the joints may be occluded for a long time (E_{colli} and E_{temp} cannot help). On the other hand, our LSTM-based predictor is trained with long time information, and thus an occluded joint could be more precisely predicted based on the long time motion information of all other joints in a hand. So we propose to use the predicted hand poses to further constrain the solution space. The prediction term $E_{\text{lstm}}(\boldsymbol{\theta}^t)$ is defined as follows:

$$E_{\text{lstm}}(\boldsymbol{\theta}^t) = \|\boldsymbol{\theta}^t - \boldsymbol{\theta}_p^t\|_2^2 \tag{11}$$

5.3.3 Energy Term for Interaction Constraint. Different from the previous terms, the interaction term (also called contact term) constrains both the motions of the hand and the object. This term is based on the simple observation that the hand will never intersect the object, which has been used in previous interaction-related papers [Tsoli and Argyros 2018]. However, due to the requirement for real-time performance, we cannot check all the surface point on the object with all the surface point on the hand. Instead, we propose a more efficient formulation.

For hand, we first sample some new spheres centered on the "skeleton" of the sphere meshes, and we only check whether an object surface point $\mathbf{v}_l$ intersects with its nearest sphere. The skeleton is a line for a type 1 sphere mesh and a triangle for a type 2 sphere mesh as shown in Fig.4, and the centers of the new spheres $\mathbf{c}$ are uniformly distributed on the line segment or in the triangle. In our implementation, we sample one new sphere for type 1, which is centered at the middle point of the line segment, and four new spheres for type 2, which are in the middle of the three edges of the triangle and the centroid of the triangle. The radius of the new spheres are also linearly interpolated by the radius of the corresponding original spheres. Then we calculate d_i to check intersection:

$$d_i(\mathbf{v}_l, \mathbf{c}) = r_c + \mathbf{n}_l(\mathbf{v}_l - \mathbf{c}). \tag{12}$$

r_c is the radius of the nearest sphere. If $d_i > 0$, $\mathbf{v}_l$ is in the sphere, otherwise, $\mathbf{v}_l$ is on or outside the sphere.

Then the interaction constraint can be formulated as:

$$E_{\text{itc}}(\mathcal{W}_{\mathcal{N}}^t, \boldsymbol{\theta}^t) = \sum_{(\mathbf{v}_l, \mathbf{c}) \in C_{\text{itc}}} \tau(d_i(\mathbf{v}_l, \mathbf{c})) * (d_i(\mathbf{v}_l, \mathbf{c}))^2 \tag{13}$$

where $\tau(r)$ is an indicator function, which equals to 1 when $r > 0$. Otherwise, it equals to 0. C_{itc} contains all the possible pairs of object surface point $\mathbf{v}_l$ and the spheres. Notice that d_i is also used to detect contact point for constructing $E_{\text{o-reg}}$. To be specific, if $d_i(\mathbf{v}_l, \mathbf{c}) > -5mm$, $\mathbf{v}_l$ is a contact point.

Notice that the interaction term works on the contact region between the fused surface of the object and the hand, so it is not related to whether the object model is fully built up or not.

5.3.4 Optimization Strategy. The minimization of the total energy of Eq.5 over the object motion field $\mathcal{W}_{\mathcal{N}}^t$ and the hand pose $\boldsymbol{\theta}^t$ is performed by a Gauss-Newton approach. We solve the motion field $\mathcal{W}_{\mathcal{N}}^t$ and $\boldsymbol{\theta}^t$ iteratively by setting the other as constant. The optimization is performed for 5 times for one frame. The contact points are updated after each iteration.

5.4 Object Model Fusion

After the tracking of both hand and object, the depth data of the object in $\{\mathcal{D}_0^t, \mathcal{D}_1^t\}$ will be fused into the canonical model $\mathcal{S}$. Notice that we could directly use the DNN-based segmentation mask of the object to identify the object depth. However, since the DNN may contain some errors and we already have a more reliable tracking result, we can use the tracked hand to further get rid of outliers. To be specific, we check whether a depth point in the object mask is close enough to the tracked hand with a threshold b (b=7mm). If so, we ignore it. Then we follow [Newcombe et al. 2015] to fuse the surface model of the object.

6 EXPERIMENTS

In this section, we first report the performance and the main parameters of the system. Then, we evaluate each of our main contributions and compare our method with previous state-of-art methods for hand tracking and object reconstruction qualitatively and quantitatively. Finally, we show the final results of our solution.

6.1 Performance and Parameters

Our system runs in real-time (running at about 40ms per frame). We use two RealSense SR300 sensors to capture hand-object interactions at 30Hz. The main algorithm for hand-object tracking and object fusion is implemented on one NVIDIA TITAN Xp GPU. The hand-object tracking takes 36ms, while the object fusion takes 2ms. Another time-consuming step includes the hand-object segmentation and the LSTM-based prediction, which take about 22ms on another GPU sever including the data streaming. Since we have made the two steps in a pipeline, we still achieve real-time performance but with a 40ms delay.

For all our experiments, we choose $\omega_{\text{itc}} = 0.2$, $\omega_{\text{d2m}} = 1$, $\omega_{\text{m2d}} = 0.5$, $\omega_{\text{pose}} = 4e2$, $\omega_{\text{lim}} = 1e7$, $\omega_{\text{colli}} = 1e3$, $\omega_{\text{temp}} = 0.05$, $\omega_{\text{lstm}} = 1e4$. The size of voxel for object fusion is 2mm in each dimension. The mean distance between neighboring nodes on object surface is 10mm. For each vertex of the object model, we find 4 nearest nodes to interpolate the motion field.

6.2 Evaluations

6.2.1 Training of LSTM based pose predictor. We collect 34 interaction sequences of one person with about 20k frames for the training

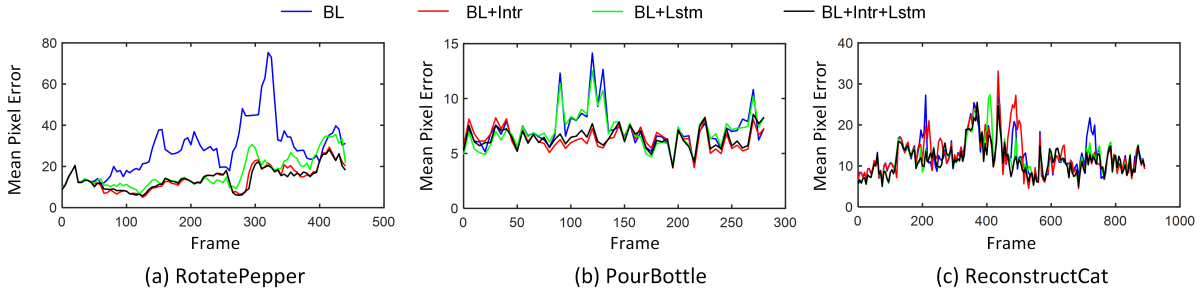

Fig. 5. Quantitative evaluations of the interaction term and the LSTM for hand tracking on different sequences. "BL" means the baseline. "Intr" means the interaction term. "Lstm" means the pose prediction term based on LSTM.

Table 1. Evaluation of the trained LSTM. To model real tracking errors, we add random noises to the inputs in the evaluation, and we calculate the errors by averaging ten epochs of evaluations for both the training set and the valuation set. Note that the error is shown in radian value, and the mean standard deviation of the selected DoFs is about 0.45.

	Train set		Evaluation set	
	All DoFs	Selected DoFs	All DoFs	Selected DoFs
Radian Errors	0.0318	0.0398	0.0399	0.0465

of our LSTM pose predictor, 16 sequences of which are about different manipulations with four real objects. We split 10% of the frames as the evaluation set. We segment our training sequences into small sub-sequences with maximum 100 frames for training, so the max unrolling steps is 100. And we set a crossover rate of 10% between two consecutive training sequences. The evaluation details are shown in Table 1 after a training of 100 epochs using Adam optimizer [Kingma and Ba 2014] with a learning rate of 0.001.

6.2.2 Ablation Study for Hand Tracking. The baseline of our method for hand tracking is [Tkach et al. 2016], which is proposed to track the isolated hands. Therefore, it faces difficulties in handling hand tracking in hand-object interactions because of the heavy occlusions and segmentation errors, even though it contains a static pose prior term, and a temporal term which considers only two previous poses. To obtain a more accurate and realistic tracking result, we introduce the interaction term and the pose prediction term based on LSTM. To quantitatively evaluate the tracking accuracy, we manually annotate a few selected keypoints on three benchmarking sequences: "RotatePepper" with 440 frames, "PourBottle" with 280 frames and "ReconstructCat" with 890 frames. We project the five tracked fingertips onto the reference color image (640x480) and measure their average errors to their ground truth. Note that we input the segmented hand of the two streams to the baseline in our experiments. Fig. 5 shows the error curves and Table 2 shows the average errors. From the results, both the interaction term and the LSTM can improve the accuracy of hand tracking, but they have different effects.

The interaction term majorly helps in handling occlusions in the input. Fig. 6 shows a frame where a pepper occludes the middle finger from both the two cameras. Adding the interaction term to the baseline method solves the wrong finger pose, because the interaction term ensures the physical plausibility, which is used as a constraint to solve the ambiguity brought by occlusions. Besides, Fig. 5 and Table 2 show the comparing results on more frames in different sequences, which sufficiently shows the power of the interaction term.

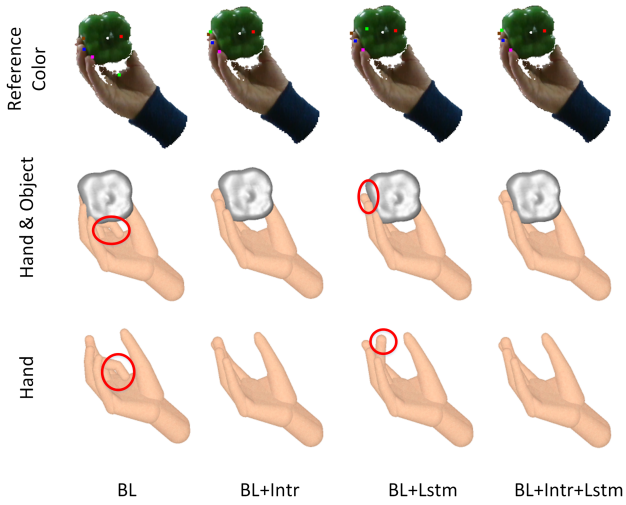

Fig. 6. Tracking results in the "RotatePepper" sequence at frame 135. The colorful points in the reference color image are the projected points of the five tracked fingertips.

Table 2. Average pixel errors of different sequences. The average is calculated from all the frames shown in Fig. 5.

	BL	BL+Intr	BL+Lstm	BL+Intr+Lstm
RotatePepper	28.4	14.4	17.1	14.1
PourBottle	7.1	6.3	7.0	6.4
ReconstructCat	12.5	12.7	11.8	11.5

The LSTM-based pose predictor has contributions in handling occlusions, segmentation errors and highly-frequency jitters. Fig. 6 demonstrates the power of LSTM in handling occlusions. Even though adding it to the baseline cannot fully solve the wrong estimation, it does improve the results a lot. Actually, the LSTM can supply a reasonable guess for the hand pose, but it cannot avoid the crossing between the hand and the reconstructed object. On the other hand, the LSTM can handle segmentation errors as shown

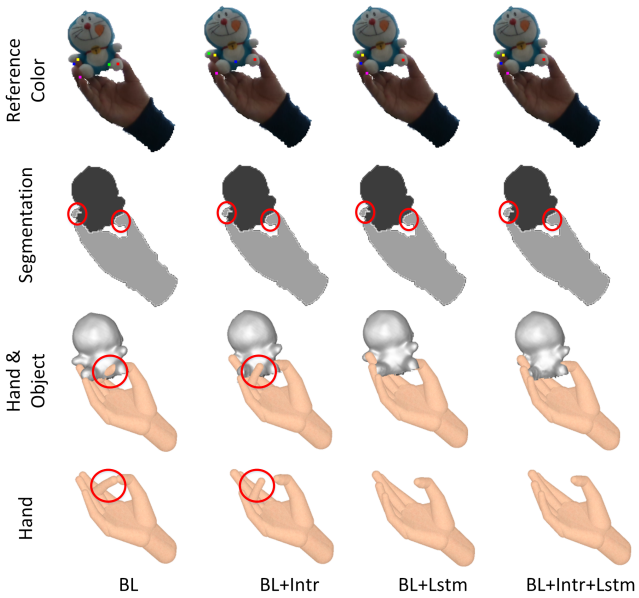

Fig. 7. Tracking results in the "ReconstructCat" sequence at frame 210.

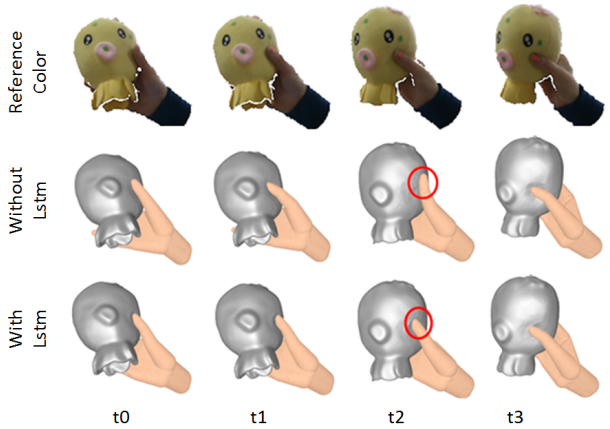

Fig. 8. Tracking results in the "Octopus" sequence at some selected frames.

in Fig. 7, where one leg and one hand of the toy is wrongly segmented as hand. Because our LSTM observes longtime information of the hand motions in interactions, it predicts reasonable poses and solves the wrong estimation. The LSTM also avoids possible highly-frequency jitters as shown in Fig. 8, because it learns to recognize the correct patterns of real hand motion sequences. The results of LSTM on more frames in more sequences are fully compared in Fig. 5 and Table 2.

6.2.3 *Ablation Study for Object Tracking.*

Spatial-temporal Varying Rigidity Regularizer for Object Tracking. In the tracking of the object, we propose a spatial-temporal varying regularizer of rigidity based on the observation that the object surface far from contact points should be more likely to be rigid, while the one close to contact points should be more likely to have non-rigid deformations. For the method using fixed rigidity coefficient, it

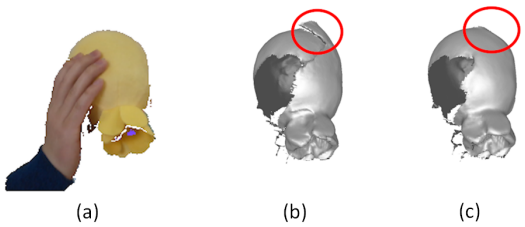

Fig. 9. Evaluation of variational rigidity strength for object loop closure. (a) Reference color. (b) Results with small rigidity strength. (c) Results with variational rigidity strength.

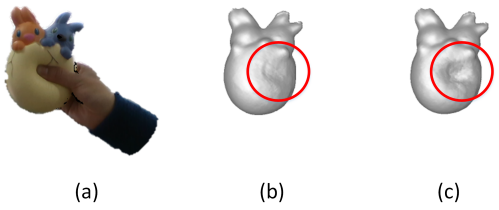

Fig. 10. Evaluation of variational rigidity strength for non-rigid motion tracking. (a) Reference color. (b) Results with big rigidity strength. (c) Results with variational rigidity strength.

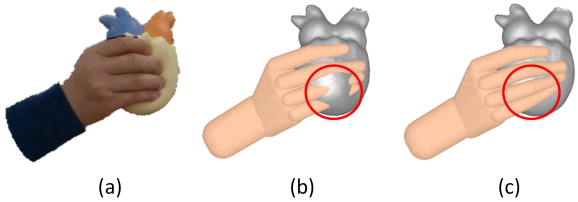

Fig. 11. Evaluation of the interaction term for object tracking. (a) Reference color. (b) Results without the interaction term. (c) Results with the interaction term.

is hard to obtain both a good loop closure (in which case we need a strong rigidity term to make the invisible regions follow the motion of visible regions) and a good non-rigid motion tracking (in which case we need a weak rigidity term to give more motion flexibility). The comparisons between fixed rigidity strength and our variational one are illustrated in Fig. 9 and Fig. 10. For a small coefficient ($\Omega(\mathbf{x})$ smaller than 40), it is hard to obtain a good rigid motion for the invisible regions, leading to a bad loop closure, which can be seen in Fig. 9(b). For a large coefficient ($\Omega(\mathbf{x})$ bigger than 30), it is hard to obtain a good estimation of nonrigid motion, which can be seen from Fig.10(b). By using variational rigidity coefficient ($\Omega(\mathbf{x})$ ranges from 5 to 60 according to the distance to contact points), a better loop closure and a better estimation of nonrigid motion can be both achieved, which is shown in Fig. 9(c) and Fig. 10(c).

Interaction Term for Object Tracking. We have introduced an interaction term in our unified optimization. For object tracking, without the interaction term, it is hard to track the deformation of the object under the press of fingers because of the occlusion, leading to unreasonable result as shown in Fig. 11(b). By introducing the interaction term, a more reasonable result is obtained, as Fig. 11(c) shows.

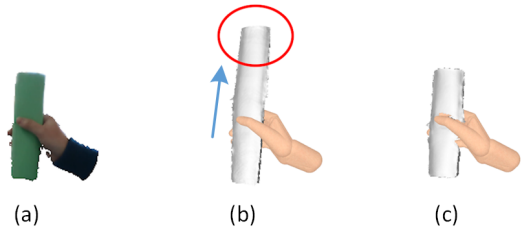

Fig. 12. Evaluation of the silhouette term for object tracking. (a) Reference color. (b) Results without the silhouette term. (c) Results with the silhouette term.

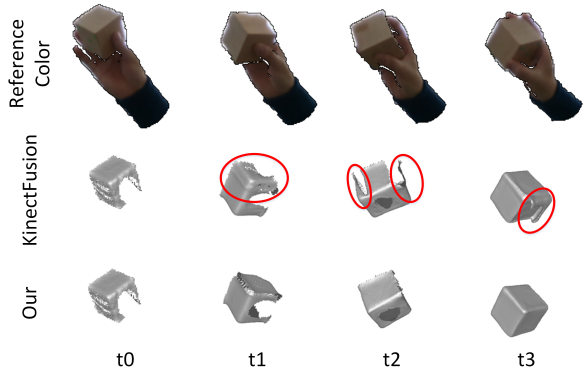

Fig. 13. Qualitative comparisons between our method and KinectFusion.

Silhouette Term for Object Tracking. We evaluate the silhouette term for object tracking in Fig. 12. For the object with fewer geometry features (for example the paper tube), the ICP-based tracking method cannot obtain accurate results. As shown in Fig. 12(b), the model of the paper tube is wrongly reconstructed to be longer than its real length. Other regularization terms cannot handle this either, i.e. neither the variational rigidity nor the interaction term can constrain the motion of the paper tube along the direction indicated by the blue arrow in Fig. 12. But after adding the silhouette term, the motion of the object can be tracked accurately as shown in Fig. 12(c).

6.3 Comparisons

We compare our reconstruction performance with KinectFusion [Newcombe et al. 2011] on a challenging sequence, rotating a cube. As shown in Fig. 13, our method obtains better fused model than KinectFusion. We also compare our tracking accuracy of the rigid object with KinectFusion on a sequence "MoveCube" quantitatively. To perform this comparison, we attach four markers on one face of the cube and measure their average projective errors on the reference color image (640x480). The ground truth of the markers is obtained by simple color thresholding. Fig. 14 shows the error curves and Fig. 15 shows the results of some frames. Our method obtains smaller tracking error than KinectFusion because we have the silhouette term and the hand-object interaction term, and the joint tracking can better solve the ambiguities of cube motions.

We compare our tracking accuracy of non-rigid motion with DynamicFusion [Newcombe et al. 2015] on a sequence "PressToy" quantitatively. We perform the comparison as that with KinectFusion, but use nine markers. Our method also obtains better tracking accuracy than DynamicFusion, as shown in Fig.16. The major reason is that for interacting motion, the occlusion is too heavy and our method introduces the interaction term to solve the ambiguities, obtaining more accurate and realistic results, as shown in Fig.17.

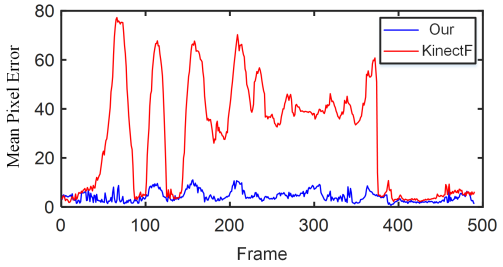

Fig. 14. Object tracking accuracy comparison with KinectFusion on sequence "MoveCube". "KinectF" means KinectFusion.

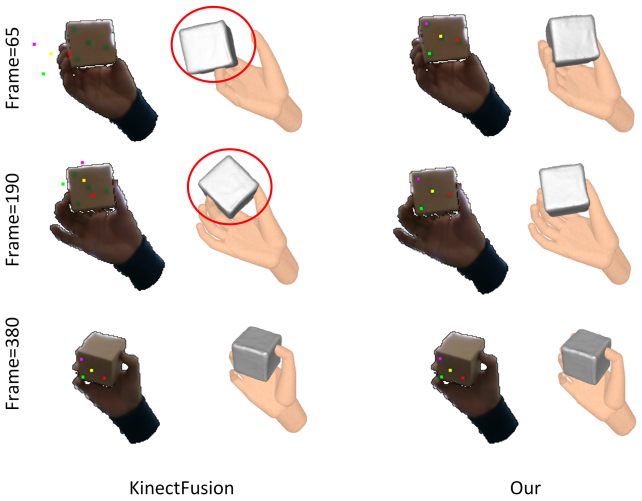

Fig. 15. Object tracking results on sequence "MoveCube". The four colorful points are the projections of the tracked vertexes.

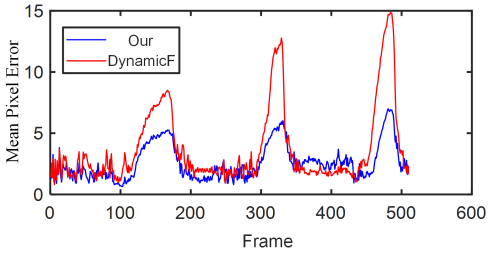

Fig. 16. Object tracking accuracy comparison with DynamicFusion on sequence "PressToy". "DynamicF" means DynamicFusion.

6.4 Results

We select some frames from our results and show them in Fig. 1 and Fig. 19. We can see that our method can handle both nonrigid and rigid objects with various shapes (from strip-like to sphere-like) and materials (paper, cloth, rigid material, elastic material, slow

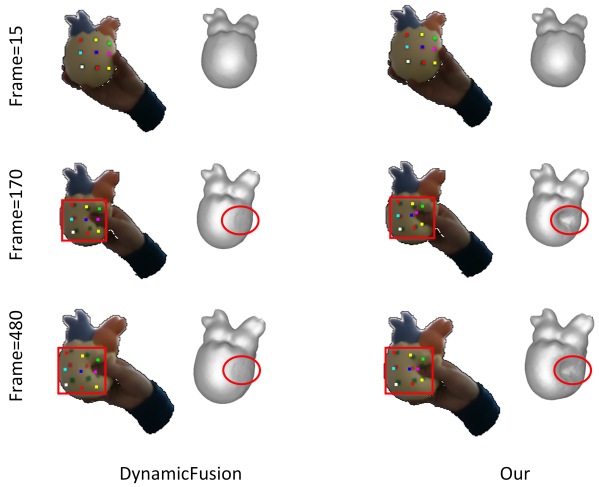

Fig. 17. Object tracking results on sequence "PressToy". The nine colorful points are the projections of the tracked vertexes.

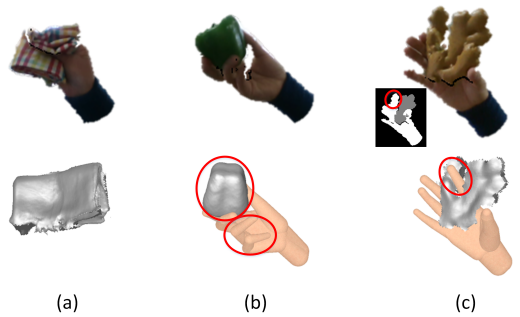

Fig. 18. Failure cases of our system. (a) Strong deformation of cloth. (b) Very fast motion. (c) Failed segmentation.

rebounding polyurethane material, and so on), and also various interactions with delicate and individual finger motions. Please refer to our accompanying videos to see more results.

6.5 Limitations and Future Work

First, we have not incorporated color information into our main system and we only use a simplified contact constraint but not a full physical regularizer which may further reduce the ambiguity. Thus there are some geometry ambiguities which we cannot solve. For example, we can only reconstruct the translation but not the rotation of a high rotational symmetric object in hand. Second, our system cannot handle some challenging cases, such as strong deformations and very fast interactions as shown in Fig.18(a, b). Since our system demands a relatively good segmentation, severe segmentation error will lead to the failure of hand tracking as shown in Fig.18(c). Finally, our system cannot handle two hands or multiple objects. We don't consider topology changes caused by interactions either. Even though our system is based on DynamicFusion [Newcombe et al. 2015], there are other great techniques which can handle topology changes of objects, such as Fusion4D [Dou et al. 2016], KillingFusion [Slavcheva et al. 2017] and SobolevFusion [Slavcheva et al. 2018]. We will refer to these techniques to explore the reconstruction of hand-object interactions with two hands, multiple objects and topology change in our future work.

7 CONCLUSIONS

This paper has proposed a novel real-time system to reconstruct hand-object interactions from two depth streams recorded at two opposite viewing points. Hand poses, 3D object models and deformations are all reconstructed in a uniform optimization framework. A DNN is used to segment the depth and an LSTM is used to predict hand poses to provide useful information. And a generative model integrates the information and further involves a spatial-temporal varying rigidity regularizer to not only help to solve ambiguities but also give enough freedom to track nonrigid motions. A sphere mesh-based contact regularizer is also proposed with a novel contact detection mechanism to guarantee the real-time performance of the system. As far as we know, this is the first system that achieves hand pose estimation, object 3D reconstruction and tracking in real time. And we believe this technique can be further used in HCI, robotics, animation and so on.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the NSFC (No.61822111, 61727808, 61671268, 61672307, 61562063) and Beijing Natural Science Foundation (L182052). Feng Xu is the corresponding author.

REFERENCES

Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. 2012. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision*. Springer, 640–653.

Zihao Bo, Hao Zhang, Junhai Yong, and Feng Xu. 2019. DenseAttentionSeg: Segment Hands from Interacted Objects Using Depth Input. *arXiv preprint arXiv:1903.12368* (2019).

Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani. 2017. Robust hand pose estimation during the interaction with an unknown object. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3123–3132.

Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. 2016. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 114.

Henning Hamer, Konrad Schindler, Esther Koller-Meier, and Luc Van Gool. 2009. Tracking a hand manipulating an object. In *Computer Vision, 2009 IEEE 12th International Conference On*. IEEE, 1475–1482.

Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D Twigg, and Kenrick Kin. 2018. Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 166.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. 2011. Manipulator and object tracking for in-hand 3D object modeling. *The International Journal of Robotics Research* 30, 11 (2011), 1311–1327.

Nikolaos Kyriazis and Antonis Argyros. 2013. Physically plausible 3d scene tracking: The single actor hypothesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9–16.

Nikolaos Kyriazis and Antonis Argyros. 2014. Scalable 3d tracking of multiple interacting objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3430–3437.

Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.

Fig. 19. Reconstruction results of our method on various materials, such as paper, cloth, rigid material, elastic materials, slow rebounding polyurethane material, and so on. For each result, we show the two reference color frames and the reconstruction results from the corresponding two viewpoints.

Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2017. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of International Conference on Computer Vision (ICCV)*, Vol. 10.

Richard A Newcombe, Dieter Fox, and Steven M Seitz. 2015. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 343–352.

Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W Fitzgibbon. 2011. Kinectfusion: Real-time dense surface mapping and tracking.. In *ISMAR*, Vol. 11. 127–136.

Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. (2011).

Paschalis Panteleris and Antonis Argyros. 2017. Back to RGB: 3D tracking of hands and hand-object interactions based on short-baseline stereo. *Hand* 2, 63 (2017), 39.

Paschalis Panteleris, Nikolaos Kyriazis, and Antonis A Argyros. 2015. 3D Tracking of Human Hands in Interaction with Unknown Objects.. In *BMVC*. 123–1.

Antoine Petit, Stéphane Cotin, Vincenzo Lippiello, and Bruno Siciliano. 2018. Capturing Deformations of Interacting Non-rigid Objects Using RGB-D Data. In *IROS 2018-IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Kha Gia Quach, Chi Nhan Duong, Khoa Luu, and Tien D Bui. 2016. Depth-based 3D hand pose tracking. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2746–2751.

Grégory Rogez, James S Supancic, and Deva Ramanan. 2015a. First-person pose recognition using egocentric workspaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4325–4333.

Grégory Rogez, James S Supancic, and Deva Ramanan. 2015b. Understanding everyday hands in action from rgb-d images. In *Proceedings of the IEEE international conference on computer vision*. 3889–3897.

Javier Romero, Hedvig Kjellström, and Danica Kragic. 2010. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 458–463.

Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. 2002. Real-time 3D model acquisition. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 438–446.

Szymon Rusinkiewicz and Marc Levoy. 2001. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 145–152.

Tanner Schmidt, Katharina Hertkorn, Richard Newcombe, Zoltan Marton, Michael Suppa, and Dieter Fox. 2015. Depth-based tracking with physical constraints for robot manipulation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 119–126.

Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. 2017. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1386–1395.

Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. 2018. SobolevFusion: 3D reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2646–2655.

Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. 2016. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision*. Springer, 294–310.

Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 143.

Jonathan Taylor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. 2017. Articulated distance fields for ultra-fast tracking of hands interacting. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 244.

Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 222.

Anastasia Tkach, Andrea Tagliasacchi, Edoardo Remelli, Mark Pauly, and Andrew Fitzgibbon. 2017. Online generative model personalization for hand tracking. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 243.

Aggeliki Tsoli and Antonis A Argyros. 2018. Joint 3D Tracking of a Deformable Object in Interaction with a Hand. In *European Conference on Computer Vision*.

Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. 2016. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision* 118, 2 (2016), 172–193.

Dimitrios Tzionas and Juergen Gall. 2015. 3d object reconstruction from hand-object interactions. In *Proceedings of the IEEE International Conference on Computer Vision*. 729–737.

Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. 2013. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 43.

Thibaut Weise, Bastian Leibe, and Luc Van Gool. 2008. Accurate and robust registration for in-hand modeling. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 1–8.

Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. 2011. Online loop closure for real-time interactive 3D scanning. *Computer Vision and Image Understanding* 115, 5 (2011), 635–648.

Carl Yuheng Ren, Victor Prisacariu, David Murray, and Ian Reid. 2013. Star3d: Simultaneous tracking and reconstruction of 3d objects using rgb-d data. In *Proceedings of the IEEE International Conference on Computer Vision*. 1561–1568.