Tracking and Reconstructing Hand Object Interactions from Point Cloud Sequences in the Wild

Jiayi Chen[1, 2*], Mi Yan[1*], Jiazhao Zhang[1], Yinzhen Xu[1, 2], Xiaolong Li[3], Yijia Weng[4], Li Yi[5], Shuran Song[6], He Wang[1†]

[1]Peking University
[2]Beijing Institute for General AI
[3]Virginia Tech
[4] Stanford University
[5]Tsinghua University
[6]Columbia University
{jiayichen, dorisyan, xuyinzhen.hi, hewang}@pku.edu.cn, {zhngjizh, halfsummer11, ericyi0124}@gmail.com, lxiaol9@vt.edu, shurans@cs.columbia.edu

Abstract

In this work, we tackle the challenging task of jointly tracking hand object pose and reconstructing their shapes from depth point cloud sequences in the wild, given the initial poses at frame 0. We for the first time propose a point cloud based hand joint tracking network, HandTrackNet, to estimate the inter-frame hand joint motion. Our HandTrackNet proposes a novel hand pose canonicalization module to ease the tracking task, yielding accurate and robust hand joint tracking. Our pipeline then reconstructs the full hand via converting the predicted hand joints into a template-based parametric hand model MANO. For object tracking, we devise a simple yet effective module that estimates the object SDF from the first frame and performs optimization-based tracking. Finally, a joint optimization step is adopted to perform joint hand and object reasoning, which alleviates the occlusion-induced ambiguity and further refines the hand pose. During training, the whole pipeline only sees purely synthetic data, which are synthesized with sufficient variations and by depth simulation for the ease of generalization. The whole pipeline is pertinent to the generalization gaps and thus directly transferable to real in-the-wild data. We evaluate our method on two real hand object interaction datasets, *e.g.* HO3D and DexYCB, without any finetuning. Our experiments demonstrate that the proposed method significantly outperforms the previous state-of-the-art depth-based hand and object pose estimation and tracking methods, running at a frame rate of 9 FPS. We have released our code on https://github.com/PKU-EPIC/HOTrack.

Introduction

Hand object interactions (HOI) are ubiquitous in our daily life and form a major way for our humans to interact with complex real-world scenes. As major approaches to perceive human object interactions, pose tracking and reconstructing human object interaction in 3D are crucial research topics and can enable a broad range of applications, including human-computer interaction, human-robot interaction, augmented reality, and learning from human demonstrations.

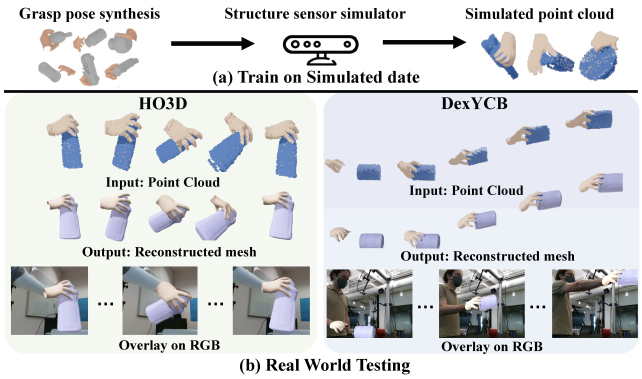

Figure 1: (a) We generate a large-scale hand-object interaction dataset, named SimGrasp, using simulated structure light based depth sensor. (b) Trained only on SimGrasp, our methods can be directly transferred to the challenging real world datasets, *i.e.* HO3D and DexYCB, to track and reconstruct hand object interaction without any finetuning.

Driven by the great power of deep learning, recent years have witnessed much progress in developing learning-based methods for perceiving hand and object under a single frame setting, *e.g.* 3D hand pose estimation from single RGB images (Xiong et al. 2019) and point clouds (Ge et al. 2018; Cheng, Park, and Ko 2021) , instance-level and category-level 6D object pose estimation (Wang et al. 2019), and joint hand object pose estimation (Doosti et al. 2020; Tekin, Bogo, and Pollefeys 2019) and reconstruction (Hasson et al. 2019; Zhang et al. 2021a) for hand object interaction.

Despite many applications require temporally coherent estimations and thus prefer tracking rather than single-frame estimation, hand object pose tracking is still an under-explored area for learning-based methods. One reason is the lack of large-scale annotated video data, given that only few

*These authors contributed equally.
†He Wang is the corresponding author.

small-scale HOI video datasets exist, e.g. DexYCB(Chao et al. 2021) and HO3D(Hampali et al. 2020a), which only cover very limited variations. Apparently, curating a large-scale fully-annotated 3D hand object interaction video dataset would be a valid step. However, the cost of doing so would be formidable, especially for 3D annotations.

In this work, we propose a novel framework to tackle this very challenging task: *jointly track pose and reconstruct the hand and object from depth point cloud sequences in the wild without seeing any real data during training*. Our setting is as follows: given a depth point cloud sequence with segmented hand and object along with initial hand pose and object pose, our system needs to recover hand object shapes and track their poses in an online manner. We choose point cloud over images since point cloud contains metric 3D geometry of the hand and object, which enables us to obtain full 3D hand and object poses and shapes, and arguably have fewer ambiguities.

To this ambitious objective, there are several major challenges. The biggest challenges come from generalization: 1) the tracking network needs to generalize well across the huge spatial and temporal variations in hand object interactions; and 2) the Sim2Real gap, due to no real training data. Also, during hand object interaction the heavy inter- and self-occlusions may result in many ambiguities and thus high ill-posedness, leading to further learning challenges.

To mitigate the data issue, we propose a simulation pipeline to synthesize diverse hand object interactions and carry free annotations of their shapes and poses. To minimize the Sim2Real gap, we leverage the structure light based depth sensor simulator proposed by DDS (Planche and Singh 2021) to generate simulated depths that carry realistic sensor noises.

Using purely the simulated data for training, we for the first time propose a point cloud based hand pose tracking network, namely HandTrackNet, to track the inter-frame hand joint motion. HandTrackNet is built upon PointNet++ (Qi et al. 2017) and can estimate the joint positions based on the predictions from the last frame. Based on the temporal continuity, it extracts the hand global pose from the latest prediction and uses it to canonicalize the hand point cloud, which significantly eases the regression task. During training, it learns to track random inter-frame joint motions and is thus free from overfitting to any temporal trajectories.

To track a novel object, we use DeepSDF (Park et al. 2019), which leverages category shape prior, to reconstruct the full geometry of the unseen object at the very first frame, and then simply perform a depth-to-SDF conformation based optimization to track pose. Though experiments we show that this simple method can already work well and allow generalization to novel object category with similar shape, *e.g.* trained on bottle and tested on box and can. Compared to previous works (Liu et al. 2021) most of which can only track known objects, we have already token a great step towards generalization.

Finally, to alleviate the ambiguities and complexity in hand object interaction, we leverage optimization-based approaches to reason the spatial relationship between the hand and object. We turn the tracked hand joint positions into a MANO (Romero, Tzionas, and Black 2017a) hand representation and construct several energy functions based on HOI priors, to yield more physically plausible hand object poses.

Our extensive experiments demonstrate the effectiveness of our method on never seen real-world hand object interaction datasets, HO3D and DexYCB. Trained only on our simulated data, our method outperforms previous approaches on both hand and object pose tracking, and shows good tracking robustness and great generalizability.

In summary, our contribution lies in three aspects: (1) We propose an online hand object pose tracking and reconstruction system taking inputs point cloud sequences under the challenging hand-object interaction scenarios, which can generalize well to never seen real-world dataset and runs at 9 FPS. (2) We propose the first point cloud-based hand pose tracking network, HandTrackNet, along with a novel hand canonicalization module, which together outperforms previous hand pose tracking or single-frame estimation methods. (3) We synthesize a large-scale diverse simulated dataset for hand-object interaction that features realistic depth sensor noise and thus enables a direct Sim2Real generalization. We will release the data and the code upon acceptance.

Related Works

3D Hand Pose Estimation and Tracking

Existing works on 3D hand pose estimation either use a 2D-3D paradigm or directly use 3D input. In the 2D-3D paradigm, for depth-based method, they either predict 2D heatmaps of hand joints (Guo et al. 2017), or regress 3D joint locations (Guo et al. 2017; Zhang and Zhang 2019). Recent works (Xiong et al. 2019; Cheng et al. 2022) explore an effective anchor-to-joint regression. RGB-based methods rely on 3D hand shape priors to lift 2D heatmaps (Zimmermann and Brox 2017; Spurr et al. 2018), or predict parameters of MANO hand model (Kulon et al. 2020; Romero, Tzionas, and Black 2017a) and hand UV maps (Chen et al. 2021). Instead of this paradigm, depth images can be transformed into 3D space, either voxels (Ge et al. 2017), point clouds (Ge et al. 2018; Cheng, Park, and Ko 2021), or T-SDF volume (Ge et al. 2017). In this work, we follow the recent trends to use point clouds as input and regress the joints in 3D space.

Early works on hand pose tracking use hand-crafted features and post optimization (Oikonomidis, Kyriazis, and Argyros 2011a) based on depth images by exploring articulation priors. When objects are present, they usually cause challenging occlusions(Mueller et al. 2017). Our method specially design to handle this occlusion.

Object Pose Estimation and Tracking

Historically, works on object pose estimation have focused on instance-level setting (Xiang et al. 2018; He et al. 2021) which only deals with known object instances. (Wang et al. 2019) extends it to a category-level setting, where poses can be estimated for novel instances from known categories. Recent works improve category-level pose estimation by leveraging shape synthesis (Chen et al. 2020), pose consistency (Lin et al. 2021), and geometrically stable patches (Huang et al. 2021)

Besides single frame pose estimation, recent works focus on temporal tracking. Instance-level object tracking approaches include optimization (Tjaden et al. 2019), filtering (Deng et al. 2019), and direct regression of inter-frame pose change (Wen et al. 2020). Recent works on category-level object tracking can extract category-level keypoints (Wang et al. 2020) without known CAD model in testing, refining coarse pose from keypoint registration by pose graph optimization (Wen and Bekris 2021), and learning inter-frame pose change from canonicalized point clouds (Weng et al. 2021). Our method goes beyond category-level pose to novel object instances from both seen and unseen categories.

Hand Object Interaction

Joint pose estimation and reconstruction of hand objects interaction has attracted much attention for its wide applications in VR, AR, teleoperation, and imitation learning (Garcia-Hernando, Johns, and Kim 2020). A number of datasets (Garcia-Hernando et al. 2018; Hasson et al. 2019; Hampali et al. 2020b,a; Chao et al. 2021) have been developed to facilitate this line of research. Many works focus on single frame pose estimation (Tekin, Bogo, and Pollefeys 2019; Hasson et al. 2020; Doosti et al. 2020; Cao et al. 2021), leveraging the interaction by feature fusion modules for joint prediction (Liu et al. 2021) or hand-object contact models (Grady et al. 2021; Yang et al. 2021) to regularize poses. We similarly adopt penetration and attraction terms to explicitly model the hand-object interaction.

Joint racking is also well explored. Early works use multi-view input to compensate for occlusions during interaction (Oikonomidis, Kyriazis, and Argyros 2011b). Recent works explore single-view tracking (Tsoli and Argyros 2018; Hasson et al. 2021) by leveraging the physical contact constraints to reconstruct plausible poses. While most of them require a known object model or a shape template, (Zhang et al. 2021a) can deal with unknown objects. During tracking, they fuse depth observations of the object in a canonical frame to progressively reconstruct the object model, which is limited to the visible surface. Our method also handles unknown objects but reconstructs the complete object from the very first frame.

Methods

Overview

Notations In this paper, we deploy the following notations: right subscripts $(\cdot)_t$ represent quantities at the time step t; and left subscripts ${}_C(\cdot)$, ${}_H(\cdot)$, ${}_O(\cdot)$ denotes variables in camera space, canonical hand space (with zero global translation and identity rotation), and canonical object spaces, respectively.

Problem setting Given a live stream of segmented hand and object point clouds$\{{}_C\mathbf{P}^{hand}, {}_C\mathbf{P}^{obj}\}_t$, along with initial hand joint positions ${}_C\mathcal{J}_{init}$ and object pose $\{\mathbf{R}^{obj}_{init}, \mathbf{T}^{obj}_{init}\}$, our aim is to recover canonical object shape ${}_O\mathcal{M}^{obj}$ and its pose $\{\mathbf{R}^{obj}_t, \mathbf{T}^{obj}_t\}$, along with hand joint positions ${}_C\mathcal{J}_t$ and hand reconstruction, including canonical hand shape ${}_H\mathcal{M}^{hand}_t$ and global hand pose

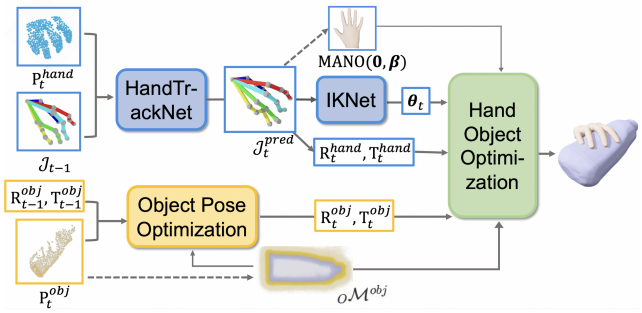

Figure 2: The full pipeline. At frame 0, We initialize the object shape ${}_O\mathcal{M}^{obj}$ represented in signed distance function (SDF) and the hand shape code $\boldsymbol{\beta}$ for the parametric model MANO, as shown in the dotted line. In the following tracking phase, at each frame t, we first separately estimate the object pose $\{\mathbf{R}^{obj}_t, \mathbf{T}^{obj}_t\}$ and hand pose $\{\mathbf{R}^{hand}_t, \mathbf{T}^{hand}_t, \boldsymbol{\theta}_t\}$, and further refine the hand pose by taking hand-object interaction into consideration.

$\{\mathbf{R}^{hand}_t, \mathbf{T}^{hand}_t\}$, for all the following frames t in an online manner. Note that the object instance under tracking can be novel, but needs to come from the training categories of our object reconstruction model , or from categories whose geometry are similar to the training categories. For hand, we only focus on the right hand here, but our method is also suitable to the left hand.

Pipeline overview. The full pipeline is shown in Fig.2. Prior to tracking, we need an initialization phase to estimate the object shape ${}_O\mathcal{M}^{obj}$ represented in SDF and the hand shape code $\boldsymbol{\beta}$ for the template-based parametric hand model MANO from the first frame.

Then, for object pose tracking, we simply perform a depth-to-SDF conformation based optimization (Zhang et al. 2021b). For hand tracking and reconstruction, we combine neural network regression and optimization approaches: at each frame t, we first leverage our point cloud based neural network, HandTrackNet, to update the hand joint positions $\mathcal{J}_t$ based on $\mathcal{J}_{t-1}$; then, to recover the full geometry of hand, we convert $\mathcal{J}_t$ into MANO hand ${}_C\mathcal{M}^{hand}_t$ with the help of IKNet; finally, we refine the hand pose based on hand-object interaction priors to recover more physical plausible interaction.

Tracking Hand Joint Positions

In this section, we introduce our point cloud based neural network, HandTrackNet, for tracking hand joint positions. During hand object interaction, many hand joints are heavily occluded, making joint position regression very challenging and ambiguous. We thus devise HandTrackNet to leverage hand joint positions $\mathcal{J}_{t-1}$ from the previous frame prediction, which distinguishes this work from all previous single-frame hand pose estimation methods (Cheng, Park, and Ko 2021; Ge et al. 2018) that directly regress $\mathcal{J}_t$ from the current observation $\mathbf{P}^{hand}_t$. We leverage $\mathcal{J}_{t-1}$ in two ways: first, $\mathcal{J}_{t-1}$ can provide a rough global pose of the hand, which

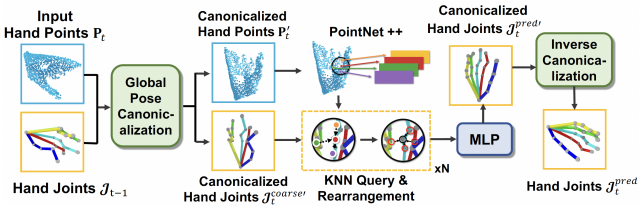

Figure 3: The architecture of HandTrackNet. HandTrackNet takes input the hand points $\mathbf{P}_t$ from the current frame t and the estimated hand joints $\mathcal{J}_{t-1}$ from the last frame, and performs global pose canonicalization to both of them. Then it leverages PointNet++ to extract features from canonicalized hand points $\mathbf{P}'_t$ and uses each joint $\mathcal{J}_t^{coarse'}$ to query and pass features, followed by an MLP to regress and update joint positions.

can be used to canonicalize the input hand point cloud $\mathbf{P}_t$, to reduce the learning complexity and increase the generalizability of different hand global poses; second, the network only needs to regress the small changes for updating $\mathcal{J}_{t-1}$, which reduces the output space of the neural network and thus eases the regression. The whole architecture of HandTrackNet is shown in Figure 3.

Hand pose canonicalization. As pointed out by Hand PointNet (Ge et al. 2018), the large variations in hand global orientation bring in significant complexity and big challenges to 3D hand pose estimation. To reduce this visual complexity, (Ge et al. 2018) proposed a point cloud canonicalization method that canonicalizes the orientation of the hand point cloud using the oriented bounding box (OBB), whose x, y, z axes are approximated by the PCA directions of input points. As a pre-processing step, this orientation canonicalization is widely adopted by the following works for single hand pose estimation, *e.g.* P2P(Ge, Ren, and Yuan 2018) and HandFoldingNet(Cheng, Park, and Ko 2021), demonstrating consistent performance improvements.

However, in the scenario of hand-object interaction, the input hand point clouds can be heavily occluded and become very partial, making PCA less effective for estimating the hand orientation and thus nullifying the effect of orientation canonicalization. To overcome this issue, we propose a simple yet novel method to estimate the 6D transformation between camera *c.s.* and hand *c.s.* from the joint positions $\mathcal{J}_{t-1}$ and then use it for canonicalization.

We start from a common assumption (Yuan et al. 2017) for 3D hand pose fitting – the wrist and 5 metacarpophalangeal (MCP) joints (where the finger bones meet the hand bones) move together and thus are fixed relative to each other. We denote this set of joints as B and their spatial positions as $\mathcal{J}^B = \{\mathcal{J}^j \mid j \in B\}$. We can further define a function $CanonPose$ (in short CP) to solve the rigid transformation for canonicalization:

$$CP(\boldsymbol{\beta},\ {}_C\mathcal{J}^B) \triangleq \arg\min_{\mathbf{R},\mathbf{T}} \|{}_H\mathcal{J}_M^B(\boldsymbol{\beta}) - \mathbf{R}^{-1}({}_C\mathcal{J}^B - \mathbf{T})\|^2 \tag{1}$$

in which ${}_H\mathcal{J}_M(\boldsymbol{\beta})$ denotes the joint positions of the template-based parametric hand model MANO, with zero translation and identity rotation. Note that ${}_H\mathcal{J}_M^B$ is irrelevant to the pose vector $\boldsymbol{\theta}$ because the bending of the fingers won't influence the position of B. The shape code $\boldsymbol{\beta}$ is initialized at frame 0. This argmin function can be analytically solved by SVD.

Finally we get $\mathbf{R}_t^B, \mathbf{T}_t^B = CanonPose(\boldsymbol{\beta},\ {}_C\mathcal{J}_{t-1}^B)$ and use it to canonicalize the hand point cloud $\mathbf{P}'_t = (\mathbf{R}_t^B)^{-1}(\mathbf{P}_t - \mathbf{T}_t^B)$. Due to temporal continuity, $\mathbf{P}'_t$ only has a small rotation and translation, which simplifies the network input and narrows down the regression output space, reducing the difficulty of regression learning.

Network architecture. After canonicalizing $\mathbf{P}_t$ to $\mathbf{P}'_t$, a naive solution is using a PointNet++ to directly regress the hand joint positions $\mathcal{J}'_t$. However, under the heavy self- and inter- occlusion, the positions of invisible joints may still be ambiguous. To alleviate this issue and ensure the joints won't go arbitrarily far, we propose to leverage the estimated joint positions from the last frame to provide a coarse estimation $\mathcal{J}_t^{coarse'} = (\mathbf{R}_t^B)^{-1}(\mathcal{J}_{t-1} + \bar{\mathbf{P}}_t - \bar{\mathbf{P}}_{t-1} - \mathbf{T}_t^B)$, which means we add the center shift between $\mathbf{P}_t$ and $\mathbf{P}_{t-1}$ to $\mathcal{J}_{t-1}$ and then transform it to the canonical frame. Now, to obtain $\mathcal{J}'_t$, the network will only need to regress a small movement for each joint and add them to $\mathcal{J}_t^{coarse'}$.

Our network first extracts the per-point feature of $\mathbf{P}'_t$ using PointNet++. Then, for each hand joint of $\mathcal{J}_t^{coarse'}$, we find k nearest neighbors in $\mathbf{P}'_t$, and aggregate those neighbors' features by a PointNet-based structure to encode the local information of $\mathcal{J}_t^{coarse'}$. To further enlarge the receptive field and encode the global information about each joint, we add a rearrangement layer to communicate among joints by concatenating the feature of adjacent joints, which is inspired by HandFoldingNet(Cheng, Park, and Ko 2021). Our rearrangement module is different from theirs because they only communicate among joints in the same finger but we also communicate with the neighbor finger. This design can provide more global information of a hand and avoid confusing the joints of different fingers, as shown in our supplementary. The above two steps, named KNN query and rearrangement, are repeated N times (we set N=2 in our paper, and we investigate different Ns in supp.) to improve the network expressivity. Finally, we use the feature of each hand joint to regress the relative movement $\mathcal{J}'_t - \mathcal{J}_t^{coarse'}$, and transform the $\mathcal{J}'_t$ back to the camera *c.s.* using the inverse transformation of canonicalization.

Training strategy To train HandTrackNet, we use single frame data and randomly perturb the ground truth hand pose by adding a Gaussian noise(we set std=2cm). Our training loss is designed as $\mathcal{L} = \lambda_{joints}\|\mathcal{J}_{gt} - \mathcal{J}_{pred}\|_1 + \lambda_{rot}\|\mathbf{R}_{gt} - \mathbf{R}_{pred}\|_1 + \lambda_{trans}\|\mathbf{T}_{gt} - \mathbf{T}_{pred}\|_1$, where $\mathbf{R}_{pred}, \mathbf{T}_{pred} = CanonPose(\boldsymbol{\beta},\ {}_C\mathcal{J}_{pred}^B)$ and they are supervised to ensure a good canonicalization for the next frame in testing. We set $\lambda_{joints} = 10$ and $\lambda_{rot} = \lambda_{trans} = 1$.

MANO Hand Reconstruction

During hand object interaction, the kinematic chain of a hand will always remain the same and only joint states may

change. Although our HandTrackNet has taken the temporal continuity into account, there is no guarantee that the hand joints will always remain the same structure, which may lead to artifacts, *e.g.* implausible joint positions and inconsistent bone length. Furthermore, without a hand shape, joint positions alone are not sufficient to investigate the interaction between the hand and the object. Thus, we propose to reconstruct the hand shape ${}_C\mathcal{M}_t^{hand}$ at each frame.

Parametric hand model. We use MANO (Romero, Tzionas, and Black 2017b), a popular template-based parametric layer, as our hand model. It maps a shape vector $\boldsymbol{\beta} \in \mathbb{R}^{10}$ and a pose vector $\boldsymbol{\theta} \in \mathbb{R}^{45}$ to a mesh ${}_H\mathcal{M}^{hand}$ and the corresponding joint positions ${}_H\mathcal{J}$. Specifically, $\boldsymbol{\beta}$ is the coefficients of learned shape PCA bases, and $\boldsymbol{\theta}$ represents the rotation of 15 joints in axis-angle representation. See (Romero, Tzionas, and Black 2017b) for more details.

Shape code initialization. Since bone lengths (or the distances between two neighboring joints) keep unchanged regardless of the joint angle $\boldsymbol{\theta}$, we can optimize the shape code $\boldsymbol{\beta}$ by minimizing

$$E(\boldsymbol{\beta}) = \|l({}_C\mathcal{J}_{pred}) - l({}_H\mathcal{J}_{MANO}(\mathbf{0}, \boldsymbol{\beta}))\|_1 \quad (2)$$

in which $l(\mathcal{J}) \in \mathbb{R}^{15}$ means the length of 15 finger bones of joints $\mathcal{J}$, and ${}_C\mathcal{J}_{pred}$ is the prediction of HandTrackNet. Note that we only estimate the shape code $\boldsymbol{\beta}$ at frame 0 here for simplicity and consistency, discussion about updating $\boldsymbol{\beta}$ can be found in Supp.

Inverse kinematic network (IKNet). To obtain joint angles $\boldsymbol{\theta}_t$, we devise a simple MLP that takes input joint position $\mathcal{J}_t'$ and output joint angles $\boldsymbol{\theta}_t$, which indeed does the job of inverse kinematics. Such a network can achieve a very high speed while also remain good performance, which suits our need well to serve as a better initialization than $\boldsymbol{\theta}_{t-1}$ for the following hand-object reasoning.

The difference between ours and (Zhou et al. 2020) is that we use the same canonicalization methods as HandTrackNet, which ensure the generalization ability (see our supp. for experiments). Specifically, we solve the global hand pose $\mathbf{R}_t^{hand}, \mathbf{T}_t^{hand} = CanonPose(\boldsymbol{\beta},\ {}_C\mathcal{J}_{pred}^B)$ and canonicalize the joint position $\mathcal{J}_t' = (\mathbf{R}_t^{hand})^{-1}(\mathcal{J}_t - \mathbf{T}_t^{hand})$, which serves as the direct input to the MLP. Finally, we can reconstuct the hand shape ${}_C\mathcal{M}_t^{hand} = \mathbf{R}_t^{hand}\ {}_H\mathcal{M}_{MANO}(\boldsymbol{\theta}_t, \boldsymbol{\beta}) + \mathbf{T}_t^{hand}$.

Object Reconstruction and Pose Tracking

In contrast to (Zhang et al. 2021a) who use a fusion-based method to reconstruct the novel object and may fail to recover the full geometry if some places are always occluded, leading to further failure in the following hand object reasoning, we leverage category shape prior to recover the full object shape at the first frame. Specifically, we propose to first estimate the signed distance function (SDF) of the object from the first frame, and then minimize a depth-to-SDF conformance (Zhang et al. 2021b) based energy function to optimize pose. SDF (Park et al. 2019) is an implicit 3D representation, which maps a spatial coordinate $\mathbf{x}$ to its signed distance s to the closest surface, *i.e.* $SDF(\mathbf{x}) = s\ :\ \mathbf{x} \in \mathbb{R}^3, s \in \mathbb{R}$.

Shape initialization. We train a DeepSDF (Park et al. 2019) using synthetic objects in ShapeNet(Chang et al. 2015). During testing, at frame 0, given the real partial point cloud of an object, we first use the given initial object pose to canonicalize this point cloud and then optimize the shape code of DeepSDF by minimizing the mean square loss of the SDF values of the depth points. Following (Park et al. 2019), we also use a L2 regularization on the shape code to alleviate the affect of the noisy real point clouds.

Object pose optimization. At frame t, given ${}_C\mathbf{P}_t^{obj}$, together with the signed distance field $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}$ of the reconstructed object shape ${}_O\mathcal{M}^{obj}$, we optimize $\{\mathbf{R}_t^{obj}, \mathbf{T}_t^{obj}\}$ to minimize

$$E(\mathbf{R}_t^{obj}, \mathbf{T}_t^{obj}) = \sum_{\mathbf{x} \in {}_C\mathbf{P}_t} \left| \psi[(\mathbf{R}_t^{obj})^{-1}(\mathbf{x} - \mathbf{T}_t^{obj})] \right| \quad (3)$$

For the choice of optimizer, we empirically find the gradient-based optimizer, *e.g.* Adam and LM, both suffer from a low speed. Therefore, we use a recently proposed gradient-free optimization method (Zhang et al. 2021b) based on random optimization. By taking the advantage of its highly parallelized framework, our method can efficiently converge to a good local optimum at a frame rate of 29 FPS under PyTorch. Please see our Supp. for more details about the comparison of different optimizers.

Shape update. In experiments, we find that our method can already generalize well to the object from a novel categories, *e.g.* trained on bottle and tested on box and can. To further mitigate the shape gap cross category and enhance the generalization ability, we also combine our method with those fusion-based method to further update the object shape during tracking. Please see our supp. for more details.

Hand-Object Optimization

Separately estimating the pose of hand and object often suffers from unrealistic hand-object interactions, as discussed in lots of related works (Hasson et al. 2019, 2021). Based on their works, we adopt the energy terms to refine the hand pose $\{\boldsymbol{\theta}, \mathbf{R}^{hand}, \mathbf{T}^{hand}\}$ by optimization at each frame.

Here we focus on using objects to help hand pose because the hand is a high DoF articulated object and the poses of those invisible fingers are very ambiguous, which is hard for HandTrackNet to estimate but can be refined using certain physical constraints with the object. In contrast, the freedom and ambiguity of object pose are much smaller, which relieves the need from hand. In our experiment (see our supp.), the error of object pose will even increase if we jointly optimize the hand pose and object pose. In the following, we will briefly introduce the energies used in this stage.

Penetration and attraction energy. Two commonly used energies for physical constraint, as proposed in (Hasson et al. 2019), are to punish hand-object penetration and encourage the contact between certain hand regions and object. However, unlike previous works (Hasson et al. 2019; Cao et al. 2021) which only focus on the static in-contact hand object interactions, for general hand object interaction

we should also consider when the hand and object are not in contact. We propose a simple strategy to decide whether to use the attraction energy by the magnitude of penetration energy. See our supplementary for details.

Joint position energy. To maintain the accuracy of joint positions, we enforce an energy to punish the L2 distance between $\mathbf{R}^{hand}\ {}_H\mathcal{J}_{MANO}(\boldsymbol{\theta}) + \mathbf{T}^{hand}$ and ${}_C\mathcal{J}_{pred}$. Since the invisible joint positions predicted by HandTrackNet may suffer from ambiguity and have a large error, this energy is only used for the visible joints. The visibility of joint i is judged by whether the distance of $\mathcal{J}_i^{pred}$ to $\mathbf{P}^{hand}$ is larger than a threshold λ ($\lambda = 2$cm in all of our experiments).

Silhouette energy. To ensure the consistency between prediction and observation, we project the reconstructed hand shape to the image plane and add a silhouette loss to punish the points that are outside the silhouette of hand and object.

Training Data Generation via Depth Simulation

To avoid using expensive real world annotation for training HandTrackNet, we propose to synthesize a large-scale point cloud dataset for hand and object interaction which we call the SimGrasp dataset. Our aim is to generate realistic point cloud sequences for diverse hand object interactions with randomized object shapes, hand shapes, grasping poses, and motion trajectories. We introduce the details of our data generation method in supplementary material. As a result, SimGrasp contains 666 different object instances and 100 hand shapes, which forms 1810 videos with 100 frames per video.

To mitigate the domain gap between synthetic depths and real depths, we re-implement a structure light based depth sensor simulator, DDS(Planche and Singh 2021), for depth image synthesis so that our simulated point clouds can capture realistic noises and artifacts. This depth simulation technique along with our fully domain randomized hand object interaction synthesis enables a direct Sim2Real transfer and allows us to test our HandTrackNet on real data in the wild.

Experiments

Datasets and Metrics

To evaluate the performance of our method on the in-the-wild data, we use two popular real world hand-object interaction datasets, HO3D(Hampali et al. 2020b) and DexYCB (Chao et al. 2021), whose hands, object instances, and motion trajectories are never seen during training. HO3D dataset is a widely used RGB-D video dataset, and DexYCB dataset is a more challenging dataset because of the larger diversity and faster motion. For a fair comparison with previous methods, we train all the methods purely on SimGrasp and directly test HO3D and DexYCB. Please see our supp. for more details about these three datasets.

We report the following metrics to evaluate the results. 1) **MPJPE**: we use the mean per joint position error for evaluating the hand pose. 2) **Penetration depth (PD)**: following (Yang et al. 2021), we report maximum penetration depth between hand and object to measure the physical plausibility of the reconstructed hand and object. 3) **Disjointedness distance (DD)**: following (Yang et al. 2021), we report the mean distance of hand vertices in 5 finger regions to their closest object vertices in frames when the ground truth hand and object are in contact. 4) **5°5cm, 10°10cm**: we follow (Weng et al. 2021) to report the 5°5cm accuracy of object pose, which means the percentage of pose predictions with rotation error $< 5°$ and translation error $<$ 5cm. 10°10cm is defined similarly. 5) **Chamfer distance (CD)**: to measure the error of the posed object geometry during tracking, we report an adapted chamfer distance $CD({}_C\mathcal{M}_{pred}^{obj}, {}_C\mathcal{M}_{gt}^{obj})$, in which $CD(\mathcal{M}_1, \mathcal{M}_2) = \frac{1}{|\mathcal{M}_1|}\sum_{\mathbf{x}\in\mathcal{M}_1}\min_{\mathbf{y}\in\mathcal{M}_2}\|\mathbf{x}-\mathbf{y}\|_2 + \frac{1}{|\mathcal{M}_2|}\sum_{\mathbf{y}\in\mathcal{M}_2}\min_{\mathbf{x}\in\mathcal{M}_1}\|\mathbf{x}-\mathbf{y}\|_2$.

	HO3D		DexYCB	
	MPJPE	PD, DD	MPJPE	PD, DD
Forth	4.04	-	4.19	-
HandFoldingNet	2.93	-	3.78	-
A2J	4.03	-	3.52	-
VirtualView	2.73	-	3.05	-
HandTrackNet	2.11	-	2.75	-
w/o hand-obj opt.	2.34	1.4, 1.5	2.99	1.8, 1.8
w/ hand-obj opt.	**2.06**	**1.1**, **1.2**	**2.69**	**1.5**, **1.4**

Table 1: Hand pose tracking. The last three lines are ours. All the methods are trained on SimGrasp and directly tested on real world dataset. '-' denotes the method doesn't reconstruct the hand shape thus can't be evaluated for those metrics. All metrics are reported in cm.

Results of Hand Pose Tracking and Reconstruction

In Table 1, we compare our method against state-of-the-art single-frame point cloud-based method HandFoldingNet (Cheng, Park, and Ko 2021), depth-based method VirtualView (Cheng et al. 2022) and A2J (Xiong et al. 2019), as well as an optimization-based tracking method Forth (Oikonomidis, Kyriazis, and Argyros 2011a). All the methods are trained on our SimGrasp and tested on real datasets without any finetune. Our method shows promising results on real world datasets, which demonstrates the generalization ability and robustness to domain gap.

In addition to predicting hand joint positions, our method also reconstructs the hand mesh. We observe that although the joint error actually increases after using the inverse kinematic network to convert joint positions into the MANO hand mesh, the following hand object optimization module reduces this error and achieves the best performance among all methods on the two real datasets. For the penetration depth and the disjointedness distance, they benefit from optimization as well, dropping by about 3mm on both real datasets. Apart from quantitative results, the visual quality of our reconstruction is also improved (see our Supp.).

Object Pose Tracking and Reconstruction

Since the object mesh is unknown for the sequences in the wild, instance-level object pose tracking methods aren't suitable for this scenario. Therefore we choose the state-of-the-art point cloud based category-level object pose track-

	HO3D		DexYCB			
	bottle	box	bottle	bowl†	box	can†
5°5cm(%)	15.7	**67.3**	16.9	23.5	23.6	22.2
10°10cm(%)	48.5	**91.9**	42.3	54.1	46.5	55.0
CD(cm)	2.84	2.22	3.97	4.97	3.83	4.93
5°5cm(%)	**43.7**	54.5	**38.5**	**33.3**	**31.2**	**35.7**
10°10cm(%)	**68.9**	84.6	**64.1**	**58.4**	**52.0**	**59.0**
CD(cm)	**2.06**	**1.74**	**2.46**	**4.71**	**2.82**	**2.32**

Table 2: Object pose tracking. The first three lines are the result of CAPTRA and the last three are ours. '†' denotes that the object category has symmetry and its rotation is evaluated by the error of symmetric axis direction.

ing method CAPTRA(Weng et al. 2021) for a comparison. However, category-level methods will also meet the situation where the testing object category shares no overlap with the training data, which creates a new challenge for the generalization ability of the existing methods.

In our experiments, we use the most similar category in SimGrasp to train CAPTRA and our method for evaluation. Note that since CAPTRA don't reconstruct the object, we use their pose to transform our reconstruction ${}_O\mathcal{M}^{obj}$ to ${}_C\mathcal{M}^{obj}$ for evaluation. See supp. for more details.

The results are reported in Table 2. We can see that though CAPTRA achieves an impressing result on SimGrasp, it fails to generalize in most cases. For HO3D, CAPTRA can do well for the object pose of the boxes, but it has a larger Chamfer distance error than ours and fails in the bottles which have a larger inter-category gap to the training objects (*i.e.* car). For DexYCB, the distortion of the point cloud caused by fast motion enlarges the data distribution gap, resulting in the bad performance of CAPTRA. In contrast, our method generalizes well in all the settings.

Ablation Study

We first verify our stereo-based depth simulation. By replacing the simulated depth with perfect depth, we generate a clean synthetic dataset for training. Figure 3 shows that the performance drops a lot on all three datasets, showing that our simulated sensor greatly reduces the Sim2Real gap.

Then, we examine our canonicalization module. If we simply remove this module or replace it with the preprocessing method OBB, the performance will drop a lot, especially on real dataset, indicating that the canonicalization module greatly improves the generalization ability.

MPJPE(cm)	SimGrasp	HO3D	DexYCB
w/o stereo depth simulation	2.53	2.49	3.68
w/o canonicalization	1.30	4.73	10.9
use OBB for canon.	2.83	4.95	4.00
w/o $\mathcal{J}_{t-1}$	0.93	2.46	2.95
w/o rearrange intra-fingers	1.16	2.69	2.82
HandTrackNet	**0.84**	**2.11**	**2.75**

Table 3: Ablation study on HandTrackNet.

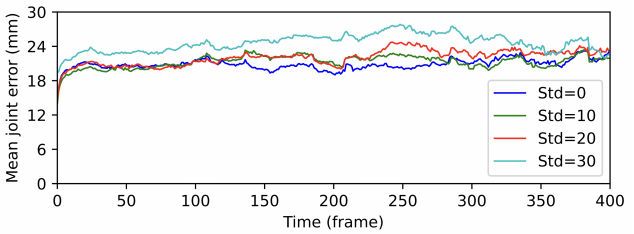

Figure 4: Robust analysis. We show the mean joint error along time on HO3D dataset with different initialization noise of $\mathcal{J}_{init}$.

We also show the importance of the last frame's joint positions $\mathcal{J}_{t-1}$ by replacing $\mathcal{J}^{coarse'}$ with a coarse estimation predicted by the bottleneck feature of PointNet++. We observe that the performance drops mainly comes from the temporal inconsistency under occlusion. Finally, we test the rearrangement module used in HandFoldingNet, which doesn't communicate joint features among different fingers. Without the information from the neighbor fingers, the network becomes easily confused by the joints on the different fingers during tracking. Some typical failure cases are shown in our supplementary materials.

Robustness and Speed

Robustness. We analyze the robustness of our methods against the noisy inputs $\mathcal{J}_{init}$ on HO3D. The initial joint position errors are randomly sampled from Gaussian distributions. We show the results with different standard deviations from 10 to 30mm in Figure 4. We can see that our method is robust to the initialization with some reasonable noise and can keep the joint error at a low level along time.

Tracking speed. Our full pipeline runs at 9 FPS in PyTorch without any C++ or CUDA acceleration tricks on an RTX 2080 Ti GPU. The HandTrackNet and object tracking module runs at 26 and 29 FPS respectively, which can be paralleled. The IKNet can reach 50FPS and the final hand-object optimization runs at 19FPS. It takes 1.15 second to initialize the object and hand shape.

Conclusions

In this paper, we present a hand-object tracking and reconstruction system with powerful generalization capability. Through extensive experiments, our method shows state-of-the-art performance on real world datasets. In the future, we would like to speed up the system by CUDA acceleration and further turn the full pipeline into an end-to-end manner.

Acknowledgements

This work is supported in part by the National Key R&D Program of China (2022ZD0114900) and the Beijing Municipal Science & Technology Commission (Z221100003422004).

References

Cao, Z.; Radosavovic, I.; Kanazawa, A.; and Malik, J. 2021. Reconstructing Hand-Object Interactions in the Wild. In *ICCV*.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chao, Y.-W.; Yang, W.; Xiang, Y.; Molchanov, P.; Handa, A.; Tremblay, J.; Narang, Y. S.; Van Wyk, K.; Iqbal, U.; Birchfield, S.; et al. 2021. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9044–9053.

Chen, D.; Li, J.; Wang, Z.; and Xu, K. 2020. Learning canonical shape space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11973–11982.

Chen, P.; Chen, Y.; Yang, D.; Wu, F.; Li, Q.; Xia, Q.; and Tan, Y. 2021. I2UV-HandNet: Image-to-UV Prediction Network for Accurate and High-Fidelity 3D Hand Mesh Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12929–12938.

Cheng, J.; Wan, Y.; Zuo, D.; Ma, C.; Gu, J.; Tan, P.; Wang, H.; Deng, X.; and Zhang, Y. 2022. Efficient Virtual View Selection for 3D Hand Pose Estimation. *arXiv preprint arXiv:2203.15458*.

Cheng, W.; Park, J. H.; and Ko, J. H. 2021. HandFoldingNet: A 3D Hand Pose Estimation Network Using Multiscale-Feature Guided Folding of a 2D Hand Skeleton. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11260–11269.

Deng, X.; Mousavian, A.; Xiang, Y.; Xia, F.; Bretl, T.; and Fox, D. 2019. PoseRBPF: A Rao-blackwellized particle filter for 6D object pose tracking. *Robotics: Science and Systems*.

Doosti, B.; Naha, S.; Mirbagheri, M.; and Crandall, D. J. 2020. HOPE-Net: A Graph-Based Model for Hand-Object Pose Estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6607–6616.

Garcia-Hernando, G.; Johns, E.; and Kim, T.-K. 2020. Physics-Based Dexterous Manipulations with Estimated Hand Poses and Residual Reinforcement Learning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9561–9568.

Garcia-Hernando, G.; Yuan, S.; Baek, S.; and Kim, T.-K. 2018. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *CVPR*.

Ge, L.; Cai, Y.; Weng, J.; and Yuan, J. 2018. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8417–8426.

Ge, L.; Liang, H.; Yuan, J.; and Thalmann, D. 2017. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1991–2000.

Ge, L.; Ren, Z.; and Yuan, J. 2018. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, 475–491.

Grady, P.; Tang, C.; Twigg, C. D.; Vo, M.; Brahmbhatt, S.; and Kemp, C. C. 2021. Contactopt: Optimizing contact to improve grasps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1471–1481.

Guo, H.; Wang, G.; Chen, X.; Zhang, C.; Qiao, F.; and Yang, H. 2017. Region ensemble network: Improving convolutional network for hand pose estimation. In *2017 IEEE International Conference on Image Processing (ICIP)*, 4512–4516. IEEE.

Hampali, S.; Rad, M.; Oberweger, M.; and Lepetit, V. 2020a. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*.

Hampali, S.; Rad, M.; Oberweger, M.; and Lepetit, V. 2020b. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3196–3206.

Hasson, Y.; Tekin, B.; Bogo, F.; Laptev, I.; Pollefeys, M.; and Schmid, C. 2020. Leveraging Photometric Consistency over Time for Sparsely Supervised Hand-Object Reconstruction. In *CVPR*.

Hasson, Y.; Varol, G.; Laptev, I.; and Schmid, C. 2021. Towards unconstrained joint hand-object reconstruction from RGB videos. In *arXiv preprint arXiv:2108.07044*.

Hasson, Y.; Varol, G.; Tzionas, D.; Kalevatykh, I.; Black, M. J.; Laptev, I.; and Schmid, C. 2019. Learning joint reconstruction of hands and manipulated objects. In *CVPR*.

He, Y.; Huang, H.; Fan, H.; Chen, Q.; and Sun, J. 2021. FFB6D: A Full Flow Bidirectional Fusion Network for 6D Pose Estimation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3002–3012.

Huang, J.; Shi, Y.; Xu, X.; Zhang, Y.; and Xu, K. 2021. StablePose: Learning 6D Object Poses from Geometrically Stable Patches. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15217–15226.

Kulon, D.; Guler, R. A.; Kokkinos, I.; Bronstein, M. M.; and Zafeiriou, S. 2020. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4990–5000.

Lin, J.; Wei, Z.; Li, Z.; Xu, S.; Jia, K.; and Li, Y. 2021. DualPoseNet: Category-level 6D Object Pose and Size Estimation Using Dual Pose Network with Refined Learning of Pose Consistency. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3540–3549.

Liu, S.; Jiang, H.; Xu, J.; Liu, S.; and Wang, X. 2021. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14687–14697.

Mueller, F.; Mehta, D.; Sotnychenko, O.; Sridhar, S.; Casas, D.; and Theobalt, C. 2017. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, 1154–1163.

Oikonomidis, I.; Kyriazis, N.; and Argyros, A. A. 2011a. Efficient model-based 3D tracking of hand articulations using Kinect. In *BmVC*, volume 1, 3.

Oikonomidis, I.; Kyriazis, N.; and Argyros, A. A. 2011b. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. *2011 International Conference on Computer Vision*, 2088–2095.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 165–174.

Planche, B.; and Singh, R. V. 2021. Physics-based Differentiable Depth Sensor Simulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14387–14397.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Romero, J.; Tzionas, D.; and Black, M. J. 2017a. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6): 1–17.

Romero, J.; Tzionas, D.; and Black, M. J. 2017b. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6).

Spurr, A.; Song, J.; Park, S.; and Hilliges, O. 2018. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 89–98.

Tekin, B.; Bogo, F.; and Pollefeys, M. 2019. H+O: Unified Egocentric Recognition of 3D Hand-Object Poses and Interactions. In *CVPR*.

Tjaden, H.; Schwanecke, U.; Schömer, E.; and Cremers, D. 2019. A Region-Based Gauss-Newton Approach to Real-Time Monocular Multiple Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 1797–1812.

Tsoli, A.; and Argyros, A. A. 2018. Joint 3D Tracking of a Deformable Object in Interaction with a Hand. In *ECCV*.

Wang, C.; Martín-Martín, R.; Xu, D.; Lv, J.; Lu, C.; Fei-Fei, L.; Savarese, S.; and Zhu, Y. 2020. 6-PACK: Category-level 6D Pose Tracker with Anchor-Based Keypoints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 10059–10066.

Wang, H.; Sridhar, S.; Huang, J.; Valentin, J.; Song, S.; and Guibas, L. J. 2019. Normalized object coordinate space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2642–2651.

Wen, B.; and Bekris, K. E. 2021. BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Wen, B.; Mitash, C.; Ren, B.; and Bekris, K. E. 2020. se(3)-TrackNet: Data-driven 6D Pose Tracking by Calibrating Image Residuals in Synthetic Domains. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Weng, Y.; Wang, H.; Zhou, Q.; Qin, Y.; Duan, Y.; Fan, Q.; Chen, B.; Su, H.; and Guibas, L. J. 2021. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13209–13218.

Xiang, Y.; Schmidt, T.; Narayanan, V.; and Fox, D. 2018. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems (RSS)*.

Xiong, F.; Zhang, B.; Xiao, Y.; Cao, Z.; Yu, T.; Zhou, J. T.; and Yuan, J. 2019. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 793–802.

Yang, L.; Zhan, X.; Li, K.; Xu, W.; Li, J.; and Lu, C. 2021. CPF: Learning a Contact Potential Field to Model the Hand-Object Interaction. In *ICCV*.

Yuan, S.; Ye, Q.; Stenger, B.; Jain, S.; and Kim, T.-K. 2017. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4866–4874.

Zhang, H.; Zhou, Y.; Tian, Y.; Yong, J.-H.; and Xu, F. 2021a. Single Depth View Based Real-Time Reconstruction of Hand-Object Interactions. *ACM Transactions on Graphics (TOG)*, 40(3): 1–12.

Zhang, J.; Zhu, C.; Zheng, L.; and Xu, K. 2021b. ROSEFusion: random optimization for online dense reconstruction under fast camera motion. *ACM Transactions on Graphics (TOG)*, 40(4): 1–17.

Zhang, X.; and Zhang, F. 2019. Pixel-wise regression: 3d hand pose estimation via spatial-form representation and differentiable decoder. *arXiv preprint arXiv:1905.02085*.

Zhou, Y.; Habermann, M.; Xu, W.; Habibie, I.; Theobalt, C.; and Xu, F. 2020. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5346–5355.

Zimmermann, C.; and Brox, T. 2017. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, 4903–4911.