HOnnotate: A method for 3D Annotation of Hand and Objects Poses

Shreyas Hampali[1], Mahdi Rad[1], Markus Oberweger[1], and Vincent Lepetit[1,2]

[1]Institute for Computer Graphics and Vision, Graz University of Technology, Austria
[2]Laboratoire Bordelais de Recherche en Informatique, Universite de Bordeaux, Bordeaux, France
{*hampali, oberweger, mahdi.rad, lepetit*}@*icg.tugraz.at*

Abstract

We propose a method for annotating images of a hand manipulating an object with the 3D poses of both the hand and the object, together with a dataset created using this method. There is a current lack of annotated real images for this problem, as estimating the 3D poses is challenging, mostly because of the mutual occlusions between the hand and the object. To tackle this challenge, we capture sequences with one or several RGB-D cameras, and jointly optimizes the 3D hand and object poses over all the frames simultaneously. This method allows us to automatically *annotate each frame with accurate estimates of the poses, despite large mutual occlusions. With this method, we created HO-3D, the first markerless dataset of color images with 3D annotations of both hand and object. This dataset is currently made of 80,000 frames, 65 sequences, 10 persons, and 10 objects, and growing, and we will make it publicly available upon publication. We also use it to train a deepnet to perform RGB-based single frame hand pose estimation and provide a baseline on our dataset.*

1. Introduction

Methods for 3D pose estimation of rigid objects and hands from monocular images have made significant progress recently, thanks to the development of Deep Learning, and the creation of large datasets or the use of synthetic images for training [51, 42, 68, 33, 29, 65]. However, these recent methods still fail when a hand interacts with an object, mostly because of large mutual occlusions, and of the absence of datasets specific to 3D pose estimation for hand+object interaction. Breaking this limit is highly desirable though, as being able to obtain accurate estimates for the hand and the object 3D poses would be very useful in augmented reality applications, or for learning by imitation in robotics, for example.

Several pioneer works have already consider this problem, sometimes with impressive success [48, 23, 56]. These works typically rely on tracking algorithms to exploit temporal constraints, often also considering physical constraints between the hand and the object for improving the pose estimates. While these temporal and physical constraints remain relevant, we would like to also benefit from the power of data-driven methods, for 3D hand+object pose estimation from a single image: Being able to estimate these poses from a single frame would avoid manual initialization and drift of tracking algorithms. A data-driven approach, however, requires real images annotated with the 3D poses of the object and the hand, or synthetic images, or both. Unfortunately, creating annotated data for the hand+object problem is very challenging. Both common options for creating 3D annotations, annotating real images and generating synthetic images, raise challenging problems:

Annotating real images. One approach is to rely on some algorithm for automated annotation, since manual annotation would be prohibitive. This is actually the approach of current benchmarks for 3D hand pose estimation [52, 41, 50, 65], where the "ground truth" annotations are obtained automatically with a tracking algorithm. These annotations are usually taken for granted and used for training and evaluation, but are actually noisy [34]. Another approach is to use sensors attached to the hand as in [12] (bottom right image of Fig. 1). This directly provides the 3D poses, however, the sensors are visible in the images, and thus bias the learning algorithm.

Generating synthetic images. Relying on synthetic images is attractive, as the 3D poses are known perfectly. Realistic rendering and domain transfer can be used to train 3D pose estimation on synthetic images [30, 43, 68]. Generating physically correct grasps is possible [27], as shown in [15]. However, real images with accurate 3D annotations would still be needed for a proper evaluation.

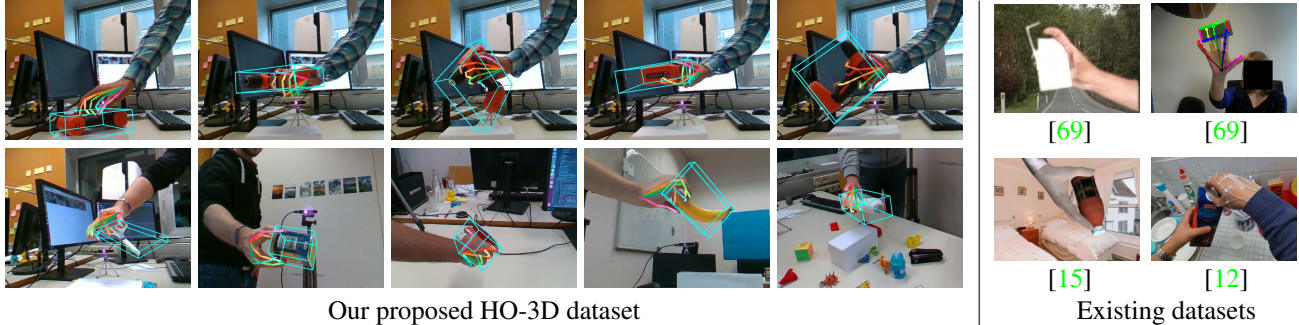We therefore propose a method to automatically annotate real images of hands grasping objects with their 3D poses.

Figure 1: We introduce a method for labelling real images of hand-object interaction with the 3D poses of the hand and of the object. With this method, we automatically created a dataset made of more than 80,000 frames, 10 different objects and 10 different users. In comparison, existing datasets have several limitations: The 3D objects are very simple, the interaction is not realistic, the images are synthetic, the images are corrupted by sensors, and/or the number of samples is limited. More illustrations of annotations in our dataset are shown in supplementary material.

Our method works with a single RGB-D camera, but can exploit more cameras if available for better robustness and accuracy. Instead of tracking the poses frame-by-frame, our method optimizes jointly all the 3D poses of the hand and the object over the sequence. As our evaluations show, this allows us to exploit temporal consistency in a way that is stronger than for tracking algorithm. Using differential rendering, we can optimize a complex objective function by exploiting the new powerful gradient descent methods originally developed for Deep Learning [21]. We see this approach as the equivalent of bundle adjustment for SLAM algorithms, where we track objects instead of points.

We rely on the MANO hand model [46], and the 3D model of the objects. We use objects from the YCB dataset [62], as they have various shapes and materials, and can be bought online [1] by researchers interested in performing their own experiments. Being able to use a single camera also enables easier expansion of the dataset by other researchers with a larger variety of objects and grasping poses as multi-camera capture is often complex to setup.

In addition to creating the HO-3D dataset of 80,000 images and our method to annotate it, we use this dataset to perform single RGB frame based hand pose estimation when interacting with objects. We train a Deep Network to predict the 2D joint locations of the hand along with the joint direction vectors and lift it to 3D by fitting a MANO model to these predictions. This validates the fact that the 3D poses estimated by our annotation method can actually be used in a data-driven method for hand pose estimation. By comparing with an existing method for hand-object pose estimation [15] that directly estimates MANO parameters, we show that predicting 2D keypoints and lifting it to 3D performs more accurately.

In the remainder of this paper, we first discuss previous work related to hand+object pose estimation. We then describe our annotation method, our dataset, and our pose esti-

mation method from a single color image. Finally, we evaluate our annotation and estimation methods.

2. Related Work

The literature on hand and/or object pose estimation is extremely broad, and we review some of the relevant works here.

2.1. 3D Object Pose Estimation

Estimating the 3D pose of an object from a single frame remains one of the fundamental problems of Computer Vision. Most methods for color images remain sensitive to occlusions, many works rely on RGB-D data to handle this problem [4, 28, 66, 19], by fitting the 3D object model to depth data. They remain sensitive to partial occlusion, especially when a hand grasps the object as the surface of the hand can be mistaken for the surface of the object.

2.2. 3D Hand Pose Estimation

Single image hand pose estimation is also a very popular problem in Computer Vision, and approaches can be divided into discriminative and generative approaches. Discriminative approaches directly predict the joint locations from RGB or RGB-D images. Recent works based on Deep Networks [54, 63, 32, 33, 29, 68, 13] show remarkable performance, compared to previous methods based on Random Forests, for example [20]. However, discriminative methods perform poorly in case of partial occlusion.

Generative approaches take advantage of a hand model and its kinematic structure to generate hand pose hypotheses that are physically plausible [49, 41, 47, 58, 26, 10, 64]. [30, 37] predict 2D joint locations and then lift them to 3D. Generative approaches are usually accurate, and can be made robust to partial occlusions. They typically rely on some prior on the hand pose, which may require manual initialization or result in drift when tracking.

Our work is related to both discriminative and generative approaches: We use a generative approach within a global optimization to generate the pose annotations, and we train a discriminative method from these data, to predict the hand and the object poses together. This way, the prediction is robust to mutual occlusions, while benefiting from the robustness of discriminative methods.

2.3. Synthetic Images for 3D Pose Estimation

Being able to train discriminative methods on synthetic data is valuable as it is difficult to acquire annotations for real images [68]. [15, 43] show that because of the domain gap between synthetic and real images, training on synthetic images only results in sub-optimal performance. A sophisticated GAN is used by [30], but this still requires renderings of high-quality synthetic color images. While using synthetic images remain attractive for many problems, creating the virtual scenes can also be expensive and time consuming. Generating animated realistic hand grasps of various objects, as it would be required to solve the problem considered in this paper remains challenging. Being able to use real sequences for training has thus also its advantages. Moreover, evaluation has to be performed on real images.

2.4. Joint Hand+Object Pose Estimation

Early approaches to joint hand-object pose estimation [35, 59, 2] typically relied on multi-view camera setups, and frame-by-frame tracking methods, which may require careful initialization and drift over time. [36, 57] propose generative methods to track finger contact points for in-hand RGB-D object shape scanning. [38, 39] consider sensing from vision to estimate contact forces during hand+object interactions using a single RGB-D camera, and then estimate the hand and the object pose. However, these methods are limited to small occlusions.

[23, 56] propose to use a physics simulator and a 3D renderer for frame-to-frame tracking of hand and objects from RGB-D. [24] uses an ensemble of Collaborative Trackers for multi-object and multiple hand tracking from RGB-D images. The accuracy of these methods seems to be qualitatively high, but as the establishment of ground truth in real-world acquisition is known to be hard, they evaluate the proposed method on synthetic datasets, or by measuring the standard deviation of the difference in hand/object poses during a grasping scenario.

[55] considers the problem of tracking a deformable object in interaction with a hand, by optimizing an energy function on the appearance and the kinematics of the hand, together with hand+object contact configurations. However, it is evaluated quantitatively only on synthetic images, which points to the difficulty of evaluation on real data. In addition, they only consider scenarios where the hand is visible from a top view, restricting the range of the hand poses and not allowing occlusions.

Very recently, [22] uses a coarse hand pose estimation to retrieve the 3D pose and shape of hand-held objects. However, they only consider a specific type of object and do not estimate the object pose. [15] presents a model with contact loss that considers physically feasible hand-object interaction to improve grasp quality. However, to estimate 3D hand pose, they predict PCA components for the pose, which results in less accurate result compared to ours, as our experiments show. [53] proposes a deep model to jointly predicts 3D hand and object poses from egocentric view, but the absence of physical constraints might result in infeasible grasps.

2.5. Hand+Object Datasets

Several datasets for hand+object interactions have already been proposed. Many works provide egocentric RGB or RGB-D sequences for action recognition [5, 6, 11, 3, 45, 25, 57]. However, they focus on grasp and action labels and do not provide 3D poses. [8, 44, 29, 55] synthetically generate datasets with 3D hand pose annotations, but fine interaction between a hand and an object remains difficult to generate accurately.

[56, 58] captured sequences in the context of hand+hand and hand+object interaction, with 2D hand annotations only. [31] collected a dataset of real RGB images of hands holding objects. They also provide 2D joint annotations of pairs of non-occluded and occluded hands, by removing the object from the grasp of the subject, while maintaining their hand in the same pose. [14] proposes two datasets, a hand+object segmentation dataset and a hand+object pose estimation dataset. However, for both datasets, the background pixels have been set to zero, and the training images only consist of a hand interacting with a tennis ball. They provide hand pose annotations and object positions, by manual labeling the joints and using a generative method to refine the joint positions. [17] generate large scale dataset with full body pose and hand pose annotation in a multi-view setup. They use a generative approach to fit the body and hand models to 3D keypoints and point cloud. However, their dataset focuses on total body pose annotation and not hand-object interactions exclusively and do not provide object pose annotations.

[48] proposed an RGB-D dataset of a hand manipulating a cube, which contains manual ground truth for both fingertip positions and 3D poses of the cube. [40] collected a dataset where they measure motion and force under different object-grasp configurations using sensors, but do not provide 3D poses. In contrast to these previous works, [12] provides a dataset of hand and object with 3D annotations for both hand joints and object pose. They used a motion capture system made of magnetic sensors attached to the

Dataset	No. of Frames	3D Object Pose	Marker-less	Real Images	Labels
PAN [17]	675K	-	+	+	automatic
GAN [30]	300K	-	+	-	synthetic
FPHA [12]	100K	-	-	+	automatic
ObMan [15]	150K	+	+	-	synthetic
Freihand [69]	37K	-	+	+	hybrid
HO-3D (ours)	80K	+	+	+	automatic

Table 1: Dataset comparison of hand+object datasets.

user s hand and to the object in order to obtain hand 3D pose annotations in RGB-D video sequences. However, this change the appearance of the hand in color images as the sensors and the tape attaching them are visible.

Very recently, [15] introduced ObMan, a large dataset of images of hands grasping objects, however, the images are synthetic and the grasps are generated using an algorithm from robotics. Even more recently, [69] proposed a multi-view RGB dataset includes hand interaction with objects, however, the annotations are limited to the 3D poses and shapes of the hand. Also, the background is synthetic, which may bias training and testing as background replacement can leave artefacts along the hand and the object's silhouettes.

As illustrated in Fig. 1 and Table 1, our HO-3D dataset is the first dataset providing both 3D hand joints and 3D object pose annotations for real images, while the hand and the object are heavily occluded by each other.

3. 3D Annotation Method

We describe below our method for annotating a sequence $\mathcal{T} = \left\{ \{(I_c^t, D_c^t)\}_{c=1}^{N_C} \right\}_{i=1}^{N_F}$ of $N_C \times N_F$ RGB-D images, captured by N_C cameras. The sequence captures a hand interacting with an object. Each RGB-D image is made of a color image I_c^t and a depth map D_c^t.

We define the 3D hand and object poses in Section 3.1, and our general cost function in Section 3.2. We initialize the poses automatically with a method described in Section 4.1, and optimize the cost function in multiple stages as described in Sections 4.2 and 4.3.

3.1. 3D Hand and Object Poses

We aim to estimate the 3D poses $\mathcal{P} = \{(p_h^t, p_o^t)\}_{t=1}^{N_F}$ for both the hand and the object in all the images of the sequence. We adopt the MANO hand model [46] and use the objects from the YCB-Video dataset [62] as their corresponding 3D models are available and of good quality. The MANO hand pose $p_h^t \in \mathbb{R}^{51}$ consists of 45 DoF (3 DoF for each of the 15 finger joints) plus 6 DoF for rotation and translation of the wrist joint. All the 16 joints (including wrist) are part of the kinematic tree with wrist joint node as the first parent node. We call the hand pose in the object coordinate system *grasp pose* in order to differentiate it from the hand pose in the world coordinate system. In addition to

the pose parameters p_h^t, the hand model has shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ that are fixed for a given person and we follow a method similar to [18] to estimate these parameters. More details about the shape parameter estimation are provided in the supplementary material. The object pose $p_o^t \in \text{SE}(3)$ consists of 6 DoF for global rotation and translation.

3.2. Cost Function

We formulate the hand-object pose estimation as an energy minimization problem:

$$\hat{\mathcal{P}} = \arg\min_{\mathcal{P}} \sum_{t=1}^{N_F} \left(E_\mathcal{D}(p_h^t, p_o^t) + E_\mathcal{C}(p_h^t, p_o^t) \right), \quad (1)$$

where $E_\mathcal{D}$ and $E_\mathcal{C}$ represent the energy from data terms and constraints, respectively. We define $E_\mathcal{D}$ as

$$\begin{aligned} E_\mathcal{D}(p_h^t, p_o^t) = \sum_{c=1}^{N_C} \Big(&\alpha E_{\text{mask}}(I_c^t, p_h^t, p_o^t) + \\ &\beta E_{\text{dpt}}(D_c^t, p_h^t, p_o^t) + \gamma E_{\text{j2d}}(I_c^t, p_h^t) \Big) + \\ &\delta E_{\text{3D}}(\{D_c^t\}_{c=1..N_C}, p_h^t, p_o^t) \,, \end{aligned} \quad (2)$$

where $E_{\text{mask}}(\cdot)$ denotes a silhouette discrepancy term, $E_{\text{dpt}}(\cdot)$ a depth residual term, $E_{\text{j2d}}(\cdot)$ denotes the 2D error in hand joint locations, and $E_{\text{3D}}(\cdot)$ a 3D distance term. This last term is not required, however, we observed that it significantly speeds up convergence. α, β, γ, and δ are weighting factors.

The constraints energy $E_\mathcal{C}$ is defined as

$$\begin{aligned} E_\mathcal{C}(p_h^t, p_o^t) = &\epsilon E_{\text{joint}}(p_h^t) + \zeta E_{\text{phy}}(p_h^t, p_o^t) + \\ &\eta E_{\text{tc}}(p_h^t, p_o^t, p_h^{t-1}, p_o^{t-1}, p_h^{t-2}, p_o^{t-2}) \,, \quad (3) \end{aligned}$$

where $E_{\text{joint}}(\cdot)$ denotes a prior on the hand pose to prevent unnatural poses, $E_{\text{phy}}(\cdot)$ is a physical plausibility term ensuring the hand and the object do not interpenetrate, and $E_{\text{tc}}(\cdot)$ is a temporal consistency term. The terms are weighted by parameters ϵ, ζ and η.

We detail each of the terms of $E_\mathcal{D}$ and $E_\mathcal{C}$ below. For simplicity, we omit the frame index t from our above notation except when necessary.

Silhouette discrepancy term E_{mask}. The $E_{\text{mask}}(\cdot)$ term compares the silhouettes of the hand and the object models rendered with the current estimated poses and their segmentation masks. We obtain a segmentation $S(I)$ of the hand and the object in color image I using DeepLabv3 [7] trained on images created by synthetically over-laying and under-laying images of hands on YCB objects. More details about this step are given in the supplementary material. The hand and object models are rendered on the camera plane using a differentiable renderer [16], which enables computing the derivatives of E_{mask} with respect to the pose parameters. The silhouette of the hand and object rendered on

camera c is denoted by $RS_c(p_h, p_o)$ and the silhouette discrepancy is defined as

$$E_{\text{mask}}(I_c, p_h, p_o) = \|RS_c(p_h, p_o) - S(I_c)\|^2 . \quad (4)$$

Depth residual term E_{dpt}. The depth residual term is similar to the segmentation discrepancy term but on the depth data:

$$E_{\text{dpt}}(D, p_h, p_o) = \text{Tukey}(\|RD_c(p_h, p_o) - D_c\|) , \quad (5)$$

where $RD_c(p_h, p_o)$ is the depth rendering of the hand and the object under their current estimated poses p_h and p_o. The Tukey function is a robust estimator that is similar to the ℓ_2 loss close to 0, and constant after a threshold. It is useful to be robust to small deviations in the scale and shape of the hand and object models and also noise in the captured depth maps. E_{dpt} is fully differentiable as we employ a differentiable renderer for rendering the depth maps.

2D Joint error term E_{j2d}. Using our initial dataset of 15,000 frames, we trained a CNN to predict the 2D locations of the 21 hand joints to further bolster our optimization strategy for the subsequent sequences. The predicted 2D joint locations in each camera image is denoted by K_c. More details on 2D joint estimation are provided in Section 4.1. The 2D joint error term is defined as

$$E_{\text{j2d}}(I_c, p_h) = \sum_{i=1}^{21} h[i]\Big\|proj_c(p_h[i]) - K_c[i]\Big\|^2, \quad (6)$$

where $p_h[i]$ denotes the 3D hand joint location under pose p_h, the $proj_c(\cdot)$ operator projects it onto camera c, $K_c[i]$ is its predicted 2D location, and $h[i]$ is the confidence for this prediction.

3D error term E_{3D}. This term is not required as the depth information from all the cameras are already exploited by E_{dpt}, however it accelerates the convergence by guiding the optimization towards the hands even from far away. We build a point cloud P by merging the depth maps from the RGB-D cameras after transforming them to a common reference frame. More details on the point cloud reconstruction can be found in the supplementary material.

We segment P into an object point cloud P_o and a hand point cloud P_h using the segmentation mask S_c in each camera image. Thus, $P = (P_o, P_h)$. At each iteration of the optimization, for each point $P_o[j]$ of the object point cloud, we look for the closest vertex $V_o[j^*]$ on the object mesh, and for each point $P_h[k]$ of the hand point cloud, we look for the closest vertex $V_h[k^*]$ on the hand mesh. $E_{\text{3D}}(P, p_h, p_o)$ is then defined as

$$\sum_j \big\|P_o[j] - V_o[j^*]\big\|^2 + \sum_k \big\|P_h[k] - V_h[k^*]\big\|^2 . \quad (7)$$

Joint angle constraint E_{joint}. This term imposes restrictions on the 15 joints of the hand to ensure the resulting pose is natural. The three dimensional rotation of a joint is parameterized using the axis-angle representation in MANO model, resulting in 45 joint angle parameters. In order to restrict the poses to natural poses, we empirically derive the limits for the 45 joint parameters (please refer supplementary material for these limits). As in [67], the joint angle constraint term $E_{\text{joint}}(p_h^t)$ is given by

$$\sum_{i=1}^{45} \max(\underline{a_i} - a[i], 0) + \max(a[i] - \overline{a_i}, 0) , \quad (8)$$

where $a[i]$ denotes the ith joint angle parameter for pose p_h, and $\underline{a_i}$ and $\overline{a_i}$ correspond to its lower and upper limits, respectively.

Physical plausibility term E_{phy}. During optimization, the hand model might penetrate the object model, which is physically not possible. To avoid this, we add a repulsion term that pushes the object and the hand apart if they interpenetrate each other. For each hand vertex $V_h[m]$, the amount of penetration $\Gamma[m]$ is taken as

$$\Gamma[m] = \max(-\mathbf{n}_o\big(V_o[m^*]\big)^T\big(V_h[m] - V_o[m^*]\big), 0) , \quad (9)$$

where $V_o[m^*]$ is the vertex on object closest to hand vertex $V_h[m]$, and the $\mathbf{n}_o(\cdot)$ operator provides the normal vector for a vertex. The physical plausibility term is defined as

$$E_{\text{phy}}(p_h^t, p_o^t) = \sum_m \exp\big(w\,\Gamma[m]\big) . \quad (10)$$

We use an exponential weight with $w = 5$ in practice, and consider only a subsampled set of vertices of the hand to compute E_{phy} efficiently.

Temporal consistency term E_{tc}. The previous terms are all applied to each frame independently. The temporal consistency term E_{tc} allows us to constrain together the poses for all the frames. We apply a 0-th and 1-st order motion model on both the hand and object poses:

$$E_{\text{tc}}(p_h^t, p_o^t, p_h^{t-1}, p_o^{t-1}, p_h^{t-2}, p_o^{t-2}) =$$
$$\|\Delta_h^t\|^2 + \|\Delta_o^t\|^2 + \|\Delta_h^t - \Delta_h^{t-1}\|^2 + \|\Delta_o^t - \Delta_o^{t-1}\|^2 ,$$

where $\Delta_h^t = p_h^t - p_h^{t-1}$ and $\Delta_o^t = p_o^t - p_o^{t-1}$. Since we optimize a sum of these terms over the sequence, this effectively constrains all the poses together.

4. Optimization

Optimizing Eq. (1) is a challenging task, as it is a highly non-convex problem with many parameters to estimate. We
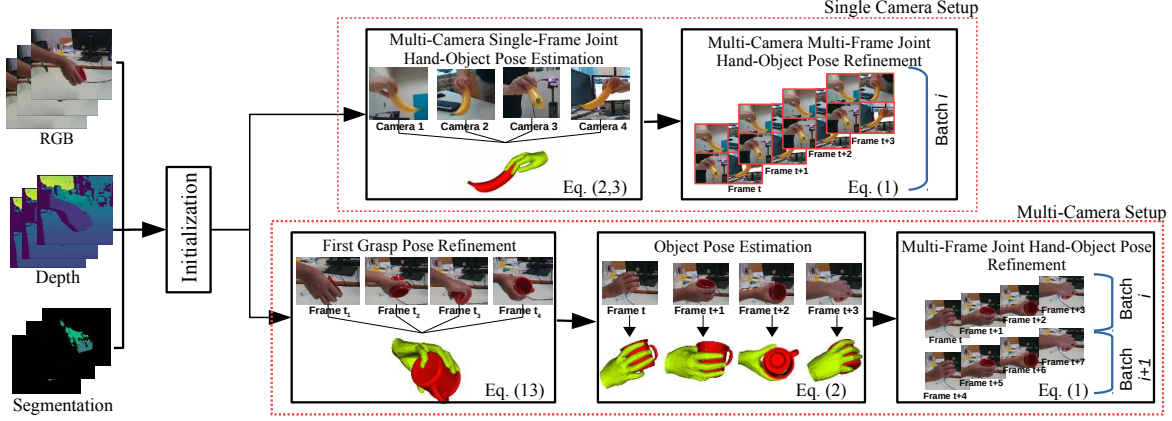
Figure 2: The different stages of the multi-camera and single camera hand-object pose annotation methods. Frame $t_{1..4}$ represent discontinuous frames in the sequence.

therefore solve the optimization problem in multiple stages as shown in Fig. 2. These stages are different for multi-camera and single camera scenarios, but in both cases, initialization of all the parameters is required. We describe our automatic initialization method and the optimization stages for both multi- and single-camera setups below.

4.1. Automatic Initialization

To perform the initialization automatically, we created a first dataset which we annotated semi-automatically, and trained a network to predict the 2D joint locations for new images. More exactly, the dataset is made of 15,000 frames from 15 sequences in a single camera setup, and we manually initialize the grasp pose and object pose for the first frame of each sequence. The manipulators were asked to maintain their grasp poses as rigid as possible to make the registration easier. We then ran the optimization stages for the single camera case described below. After optimization, we augmented the resulting dataset by scaling and rotating the images, and adding images from the Panoptic Studio dataset [61], which contain 3D annotations for hands. We used the resulting dataset to train a CNN based on the architecture of [60] to predict 2D projections of the 21 hand joints K in the image, in the form of heatmaps trained with the ℓ_2 loss function. Given a new sequence, we use the 2D joint predictions made by this CNN to initialize the hand poses as explained below.

In the multi-camera setup, the initialization for the hand pose p_h^t for frame t is taken as

$$p_h^t = \arg\min_{p_h^t} \sum_{c=1}^{N_C} E_{\text{j2d}}(I_c^t, p_h) + \nu E_{\text{joint}}(p_h), \quad (11)$$

where E_{j2d} is as defined in Eq. (6)—in this term $K_c[k]$ is taken as the location of the peak in the predicted heatmap

for joint k and $h[k]$ as the magnitude of this peak. $E_{\text{joint}}(p_h)$ is defined as in Eq. (8).

In the single camera setup, we initially make the assumption that the grasp pose of the hand does not vary during the sequence. We relax this assumption in the latter stages of the optimization, but it is reasonable as a first estimate as it is difficult to change a grasp when holding an object, especially if the object is large, which is when difficult occlusions occur. We select a random subset Ω of frames from the sequence, and we estimate the grasp pose by minimizing

$$\sum_{t\in\Omega} E_{\text{j2d}}(I^t, p_h^t) + \nu \sum_{t\in\Omega} E_{\text{joint}}(p_h^t), \quad (12)$$

where the joint angle parameters are constrained to be the same over all the frames, and only the rotation and parameters for the wrist joint can be different. We set $\nu = 50$ in our experiments and use dogleg optimizer[9] for minimizing Eq. 11 and Eq. 12.

In both setups, the object poses p_o^t are initialized by using the method from [42], trained by synthetically over-laying hands on YCB objects as explained in Section 3.2.

4.2. Multi-Camera Setup

After the pose initialization for the first frame, we obtain a first estimate for the object and hand poses for all the other frames by tracking them. This is done by minimizing Eq. (1) on one frame only, from $t = 1$ to N_F and using (p_h^{t-1}, p_o^{t-1}) to initialize (p_h^t, p_o^t). We then perform a full optimization of Eq. (1) over all the frames simultaneously as illustrated in Fig. 2. Due to memory constraints, we optimize Eq. (1) in batches instead of considering all the frames in sequence. We use a batch size of 20 frames and use the following values for the weights, $\alpha = 20, \beta = 20, \gamma = 5 \times 10^{-5}, \delta = 50, \epsilon = 100, \zeta = 50,$

and $\eta = 100$, and optimize using Adam optimizer with learning rate of 0.01 for 100 iterations.

4.3. Single Camera Setup

Using the initialization for the grasp and object poses as explained in Section 4.1, the final grasps and object poses are estimated in 3 stages as shown in Fig. 2. We first estimate the grasp pose from multiple frames in the first stage, which is then used to estimate object pose in each frame in the second stage. In the final stage, we employ temporal constraints and jointly optimize hand and object poses over multiple frames and allow for variation in grasp pose across frames. More details about these stages is provided below.

First grasp pose refinement. Starting from the first estimate for the grasp pose, we refine it a first time by minimizing a cost function closer to our final objective:

$$E(p_h, \{p_o^t\}_{t \in \Omega}) = \sum_{t \in \Omega} \Big(E_{\mathcal{D}}(p_h, p_o^t) + \zeta E_{\text{phy}}(p_h, p_o^t) \Big)$$

$$+ \epsilon E_{\text{joint}}(p_h) \,,$$

(13)

where Ω is a set of randomly selected frames, and the grasp pose defined with respect to the object is still constrained to be the same across all frames. Note that this time, the object pose parameters are also optimized, to improve our grasp pose estimate.

Object pose estimation. The hand and object meshes under the estimated grasp pose models the mutual occlusion between the hand and object. The object poses p_o^t are refined successively, by minimizing $E_{\mathcal{D}}(p_h^t, p_o^t)$ in Eq. (2), with respect to only object pose parameters p_o^t and using p_o^{t-1} to initialize p_o^t.

Joint hand-object pose refinement. In the final stage, we finally allow variations in the grasp pose and introduce temporal constraints E_{tc} for smooth variations of the poses: We optimize Eq. (1) over all the parameters, initialized with the values obtained in the previous stages.

5. Monocular RGB based 3D Hand Pose Estimation Method

For establishing a baseline on our proposed dataset for single RGB image based hand pose estimation, we use a CNN architecture based on a Convolutional Pose Machine (CPM) [60] to predict the 2D hand joint locations $\{k_i\}_{i=1..21}$ as in the automatic initialization described in Section 4.1. In addition, we also predict the root relative hand joint directions $\{d_i\}_{i=1..20}$, by adding an additional stage at the end of the CPM and replacing the last layer

with fully connected layer. More details on the architecture is provided in supplementary material. The 3D joint locations and shape parameters of the hand are then obtained by fitting a MANO model to these predictions. The loss function for this fitting procedure is given by:

$$\mathcal{L} = \sum_{i=1}^{21} \|\hat{k}_i - k_i\|^2 + \rho \sum_{i=1}^{20} \big(1 - \hat{d}_i \cdot d_i\big) + \tag{14}$$

$$\sigma E_{\text{joint}}(p_h) + \tau \|\beta\|^2 \,,$$

where $\hat{d}_i = \frac{p_h[i] - p_h[1]}{\|p_h[i] - p_h[1]\|}$, $\hat{k}_i = proj\big(p_h[i]\big)$ and E_{joint} is defined in Eq. (8). The weights ρ, σ, and τ are set to 10, 5, and 1, respectively.

6. Benchmarking the Dataset

In this section, we evaluate both our annotation method and our baseline for hand pose prediction from a single color image in hand-object interaction scenarios. We used our 3D pose annotation method to annotate 65 sequences (so far), totaling about 80,000 frames of 10 different users manipulating one among 10 different objects from the YCB dataset. The image sizes are 640×480 pixels for both the color and depth cameras, and we used 5 synchronized cameras in our multi-camera setup.

6.1. Evaluation of the Annotation Method

For validating the accuracy of our annotation method, we manually annotated the 3D locations of the 3D joints in randomly selected frames of a sequence, by relying on the consolidated point cloud from the 5 cameras. We then compared these locations to the ones predicted with our method using the multi-camera setup.

As shown on the last column of Table 2, our method achieves an accuracy lower than 8mm on average, with an Area Under the Curve metric (AUC) of 0.79. This metric is comparable with the results reported for the recent FreiHAND dataset [69].

To also analyze the influence of the different terms in Eq. (1), we run the optimization of Eq. (1) by enabling only a subset of these terms, and report the results in Table 2.

While E_{silh} and E_{dpt} terms alone cannot provide good pose estimates, together they provide better estimates as it leads to a loss function with less local minimums. The E_{3D} term provides a minor improvement in estimates but speeds up the convergence. Though the physical plausibility term E_{phy} does not help in improving the pose estimates, it results in more natural grasps. The last two columns represents the effect of introducing temporal consistency term locally and globally. The multi-frame multi-camera based optimization over all the terms improves the accuracy by about 15%.

For the single camera method, we computed the distances between the meshes for the hand and the object retrieved by the single camera setup and the multi-camera

Terms	Initialization	E_{silh}	E_{dpt}	$E_{silh}+E_{dpt}$	$E_{silh}+E_{dpt}+E_{3D}$	$E_{silh}+E_{dpt}+E_{3D}+E_{phy}$	$E_{silh}+E_{dpt}+E_{3D}+E_{phy}+E_{tc}$	Multi-frame
mean (std)	4.20 (±3.32)	1.17 (±1.12)	2.22 (±1.22)	1.04 (±0.43)	0.98 (±0.40)	0.99 (±0.40)	0.92 (±0.34)	0.77 (±0.29)

Table 2: Accuracy evaluation (in cm) and ablation study of the terms in Eq. (1) for hand in multi-camera setup. The error for the complete method is comparable to the recent FreiHAND dataset [69].

Stages	Init.	Grasp Pose Est.	Object Pose Est.	Refinement
Hand	5.40	3.60	0.91	0.77
Object	4.02	4.02	0.52	0.45

Table 3: Comparing the pose estimates at the different stages of the single camera setup to the final estimates by the multi-camera setup (average mesh distances in cm) averaged over several frames of a sequences. The results show that the estimates from two methods are close, which shows that our method achieves high accuracy even with a single RGB-D camera.

Method	Mesh Error	F@5mm	F@15mm	Joint Error
Joints2D	1.29	0.46	0.90	4.59
Joints2D + Dir. Vec.	**1.07**	**0.48**	**0.94**	**2.95**
[15]	1.15	0.47	0.93	6.23

Table 4: Evaluation of different methods for single frame based hand pose estimation. Mesh error (in cm) and *F*-score are obtained after aligning the predicted meshes with ground truth meshes. Mean joint error (in cm) is obtained after aligning the position of the root joint and overall scale with the ground truth. Hand pose estimation using joint direction (wrt wrist joint) predictions along with 2D joint predictions provides better accuracy than directly predicting MANO parameters as in [15].

setup for a sequence of 1000 frames, after each of the optimization stages. The results are given in Table 3. The estimated poses with these two methods are consistent with each other with a average mesh error of 0.77cm and 0.45cm for hand and object, respectively. The final refinement stage yields a 15% improvement in accuracy.

6.2. Evaluation of the Single Frame Hand Pose Prediction Method

We trained our single frame-based hand pose estimation method explained in Section 5 on 74,000 frames from our HO-3D dataset. We evaluated it on a test set of 7 sequences captured from different viewpoints and totaling about 7000 frames. The test set also contains manipulators not present in the training set, and 4 different objects are manipulated.

We report three different metrics from previous works: Mean joint position error after aligning the position of the root joint and global scale with ground truth [68]; Mesh error measuring the average Euclidean distance between predicted and ground truth mesh [69]; and the *F*-score [69], defined as the harmonic mean between recall and precision between two meshes given a distance threshold. The mesh
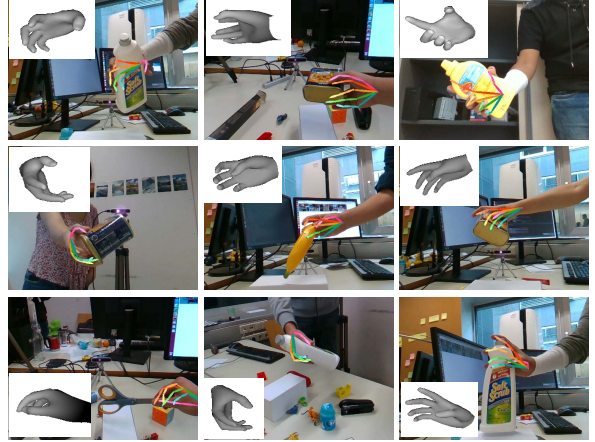

Figure 3: Qualitative results for single RGB frame based hand pose estimation method. We recover hand poses when it is heavily occluded by objects and in cluttered scenes.

error and *F*-score are obtained after aligning the predicted meshes using Procrustes alignment with the ground truth meshes and hence does not measure the accuracy of wrist joint rotation. The mean joint error on the other hand considers wrist joint location as the 3D points are not rotated before evaluation.

To understand the effect of joint direction predictions on the overall accuracy, we evaluate the results of the MANO fitting by dropping the second term in Eq. (14). We also compare our results with [15], a very recent work that predicts the MANO pose and shape parameters directly from a single RGB image, retrained on our dataset.

As shown in Table 4, predicting joint directions along with 2D joint locations significantly improves the hand pose estimation accuracy. It can also be inferred that predicting 2D hand joint locations and lifting them to 3D is more accurate than direct MANO parameter predictions as in [15]. Qualitative results are shown in Fig. 3.

7. Conclusion

We introduced a fully automatic method to 3D annotate images of a hand manipulating an object with their 3D poses. We also introduced the first markerless dataset of color images for benchmarking 3D hand+object pose estimation, together with an evaluation strategy for this method, and a baseline method for predicting the 3D pose of the hand and the object from a single color image. Our method can deal with large occlusions by exploiting temporal con-

sistency, and can probably be extended to other 3D annotation problems.

References

[1] YCB Benchmarks Object and Model Set. http://ycbbenchmarks.org/. 2

[2] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys. Motion Capture of Hands in Action Using Discriminative Salient Points. In *ECCV*, 2012. 3

[3] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In *ICCV*, 2015. 3

[4] A. G. Buch, L. Kiforenko, and D. Kraft. Rotational Subgroup Voting and Pose Clustering for Robust 3D Object Recognition. In *ICCV*, 2017. 2

[5] I. M. Bullock, T. Feix, and A. M. Dollar. The Yale Human Grasping Dataset: Grasp, Object, and Task Data in Household and Machine Shop Environments. *The International Journal of Robotics Research*, 34(3):251–255, 2015. 3

[6] M. Cai, K. M. Kitani, and Y. Sato. A Scalable Approach for Understanding the Visual Structures of Hand Grasps. In *ICRA*, 2015. 3

[7] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR*, abs/1706.05587, 2017. 4

[8] C. Choi, S. Ho Yoon, C.-N.Chen, and K. Ramani. Robust Hand Pose Estimation During the Interaction with an Unknown Object. In *ICCV*, 2017. 3

[9] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, PA, USA, 2000. 6

[10] M. de L. Gorce, D. Fleet, and N. Paragios. Model-Based 3D Hand Pose Estimation from Monocular Video. *PAMI*, 33(9):1793–1805, 2011. 2

[11] A. Fathi, X. Ren, and J. M. Rehg. Learning to Recognize Objects in Egocentric Activities. In *CVPR*, 2011. 3

[12] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In *CVPR*, 2018. 1, 2, 3, 4

[13] L. Ge, Y. Cai, J. Weng, and J. Yuan. Hand PointNet: 3D Hand Pose Estimation Using Point Sets. In *CVPR*, 2018. 2

[14] D. Goudie and A. Galata. 3D Hand-Object Pose Estimation from Depth with Convolutional Neural Networks. In *IEEE International Conference on Automatic Face & Gesture Recognition*, 2017. 3

[15] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning Joint Reconstruction of Hands and Manipulated Objects. In *CVPR*, 2019. 1, 2, 3, 4, 8

[16] P. Henderson and V. Ferrari. Learning Single-Image 3D Reconstruction by Generative Modelling of Shape, Pose and Shading. *IJCV*, 2019. 4

[17] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, pages 8320–8329, 2018. 3, 4

[18] David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4

[19] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation. In *ECCV*, 2016. 2

[20] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Hand Pose Estimation and Hand Shape Classification Using Multi-Layered Randomized Decision Forests. In *ECCV*, 2012. 2

[21] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICML*, 2015. 2

[22] M. Kokic, D. Kragic, and J. Bohg. Learning to Estimate Pose and Shape of Hand-Held Objects from RGB Images. In *arXiv Preprint*, 2019. 3

[23] N. Kyriazis and A. Argyros. Physically Plausible 3D Scene Tracking: the Single Actor Hypothesis. In *CVPR*, 2013. 1, 3

[24] N. Kyriazis and A. Argyros. Scalable 3D Tracking of Multiple Interacting Objects. In *CVPR*, 2014. 3

[25] R. Luo, O. Sener, and S. Savarese. Scene Semantic Reconstruction from Egocentric RGB-D-Thermal Videos. In *3DV*, 2017. 3

[26] S. Melax, L. Keselman, and S. Orsten. Dynamics Based 3D Skeletal Hand Tracking. In *Proceedings of Graphics Interface*, 2013. 2

[27] A. T. Miller and P. K. Allen. Graspit! a Versatile Simulator for Robotic Grasping. *Robotics Automation Magazine*, 2004. 1

[28] C. Mitash, A. Boularias, and K. E. Bekris. Improving 6D Pose Estimation of Objects in Clutter via Physics-Aware Monte Carlo Tree Search. In *ICRA*, 2018. 2

[29] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-Time Hand Tracking Under Occlusion from an Egocentric Rgb-D Sensor. In *ICCV*, 2017. 1, 2, 3

[30] F. Müller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Ganerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In *CVPR*, 2018. 1, 2, 3, 4

[31] B. Myanganbayar, C. Mata, G. Dekel, B. Katz, G. Ben-Yosef, and A. Barbu. Partially Occluded Hands: A Challenging New Dataset for Single-Image Hand Pose Estimation. In *ACCV*, 2018. 3

[32] N. Neverova, C. Wolf, F. Nebout, and G. Taylor. Hand Pose Estimation through Semi-Supervised and Weakly-Supervised Learning. *CVIU*, 2017. 2

[33] M. Oberweger and V. Lepetit. DeepPrior++: Improving Fast and Accurate 3D Hand Pose Estimation. In *ICCV*, 2017. 1, 2

[34] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit. Efficiently Creating 3D Training Data for Fine Hand Pose Estimation. In *CVPR*, 2016. 1

[35] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full DoF Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints. In *ICCV*, 2011. 3

[36] P. Panteleris, N. Kyriazis, and A. A. Argyros. 3D Tracking of Human Hands in Interaction with Unknown Objects. In *BMVC*, 2015. 3

[37] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a Single RGB Frame for Real Time 3D Hand Pose Estimation in the Wild. In *WACV*, 2018. 2

[38] T.-H. Pham, A. Kheddar, A. Qammaz, and A. Argyros. Capturing and Reproducing Hand-Object Interactions through Vision-Based Force Sensing. In *Object Understanding for Interaction*, 2015. 3

[39] T.-H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros. Towards Force Sensing from Vision: Observing Hand-Object Interactions to Infer Manipulation Forces. In *CVPR*, 2015. 3

[40] T.-H. Pham, N. Kyriazis, A. A. Argyros, and A. Kheddar. Hand-Object Contact Force Estimation from Markerless Visual Tracking. *PAMI*, 40(12):2883–2896, 2018. 3

[41] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and Robust Hand Tracking from Depth. In *CVPR*, 2014. 1, 2

[42] M. Rad and V. Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In *ICCV*, 2017. 1, 6

[43] M. Rad, M. Oberweger, and V. Lepetit. Domain Transfer for 3D Pose Estimation from Color Images Without Manual Annotations. In *ACCV*, 2018. 1, 3

[44] G. Rogez, M. Khademi, S. III, J. A. Montiel, J. M. Martinez, and D. Ramanan. 3D Hand Pose Detection in Egocentric RGB-D Images. In *ECCV*, 2014. 3

[45] G. Rogez, J. S. Supancic, and D. Ramanan. Understanding Everyday Hands in Action from RGB-D Images. In *ICCV*, 2015. 3

[46] J. Romero, D. Tzionas, and M. J. Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *TOG*, 36(6):245, 2017. 2, 4

[47] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, Robust, and Flexible Real-Time Hand Tracking. In *CHI*, 2015. 2

[48] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-Time Joint Tracking of a Hand Manipulating an Object from RGB-D Input. In *ECCV*, 2016. 1, 3

[49] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. In *ICCV*, 2013. 2

[50] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded Hand Pose Regression. In *CVPR*, 2015. 1

[51] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *ECCV*, 2018. 1

[52] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture. In *CVPR*, 2014. 1

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

[54] J. Tompson, M. Stein, Y. LeCun, and K. Perlin. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *TOG*, 33, 2014. 2

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

[56] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing Hands in Action Using Discriminative Salient Points and Physics Simulation. *IJCV*, 118(2):172–193, 2016. 1, 3

[57] D. Tzionas and J. Gall. 3D Object Reconstruction from Hand-Object Interactions. In *ICCV*, 2015. 3

[58] D. Tzionas, A. Srikantha, P. Aponte, and J. Gall. Capturing Hand Motion with an RGB-D Sensor, Fusing a Generative Model with Salient Points. In *German Conference on Pattern Recognition*, 2014. 2, 3

[59] R. Wang, S. Paris, and J. Popović. 6D Hands: Markerless Hand-Tracking for Computer Aided Design. In *ACM Symposium on User Interface Software and Technology*, 2011. 3

[60] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. In *CVPR*, 2016. 6, 7

[61] D. Xiang, H. Joo, and Y. Sheikh. Monocular Total Capture: Posing Face, Body, and Hands in the Wild. In *CVPR*, 2019. 6

[62] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *RSS*, 2018. 2, 4

[63] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng. Lie-X: Depth Image Based Articulated Object Pose Estimation, Tracking, and Action Recognition on Lie Groups. *IJCV*, 2016. 2

[64] Q. Ye, S. Yuan, and T.-K. Kim. Spatial Attention Deep Net with Partial PSO for Hierarchical Hybrid Hand Pose Estimation. In *ECCV*, 2016. 2

[65] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim. Big Hand 2.2m Benchmark: Hand Pose Data Set and State of the Art Analysis. In *CVPR*, 2017. 1

[66] H. Zhang and Q. Cao. Combined Holistic and Local Patches for Recovering 6D Object Pose. In *ICCV*, 2017. 2

[67] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-Based Deep Hand Pose Estimation. *IJCAI*, 2016. 5

[68] C. Zimmermann and T. Brox. Learning to Estimate 3D Hand Pose from Single RGB Images. In *ICCV*, 2017. 1, 2, 3, 8

[69] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images. In *ICCV*, 2019. 2, 4, 7, 8