Learning joint reconstruction of hands and manipulated objects

Yana Hasson[1,2] Gül Varol[1,2] Dimitrios Tzionas[3] Igor Kalevatykh[1,2]
Michael J. Black[3] Ivan Laptev[1,2] Cordelia Schmid[1]

[1]Inria, [2]Département d'informatique de l'ENS, CNRS, PSL Research University
[3]MPI for Intelligent Systems, Tübingen

Abstract

Estimating hand-object manipulations is essential for interpreting and imitating human actions. Previous work has made significant progress towards reconstruction of hand poses and object shapes in isolation. Yet, reconstructing hands and objects during manipulation is a more challenging task due to significant occlusions of both the hand and object. While presenting challenges, manipulations may also simplify the problem since the physics of contact restricts the space of valid hand-object configurations. For example, during manipulation, the hand and object should be in contact but not interpenetrate. In this work, we regularize the joint reconstruction of hands and objects with manipulation constraints. We present an end-to-end learnable model that exploits a novel contact loss that favors physically plausible hand-object constellations. Our approach improves grasp quality metrics over baselines, using RGB images as input. To train and evaluate the model, we also propose a new large-scale synthetic dataset, ObMan, with hand-object manipulations. We demonstrate the transferability of ObMan-trained models to real data.

1. Introduction

Accurate estimation of human hands, as well as their interactions with the physical world, is vital to better understand human actions and interactions. In particular, recovering the 3D shape of a hand is key to many applications including virtual and augmented reality, human-computer interaction, action recognition and imitation-based learning of robotic skills.

Hand analysis in images and videos has a long history in computer vision. Early work focused on hand estimation and tracking using articulated models [15, 44, 58, 71] or statistical shape models [26]. The advent of RGB-D sensors brought remarkable progress to hand pose estimation from depth images [13, 20, 36, 60, 62]. While depth sensors provide strong cues, their applicability is limited by the energy consumption and environmental constrains such as distance to the target and exposure to sunlight. Recent work obtains promising results for 2D and 3D hand pose estimation from monocular RGB images using convolutional neural networks [7, 16, 32, 40, 55, 56, 73]. Most of this work, however, targets sparse keypoint estimation which is not sufficient for reasoning about hand-object contact. Full 3D hand meshes are sometimes estimated from images by fitting a hand mesh to detected joints [40] or by tracking given a good initialization [6]. Recently, the 3D *shape* or *surface* of a hand using an end-to-end learnable model has been addressed with depth input [28].

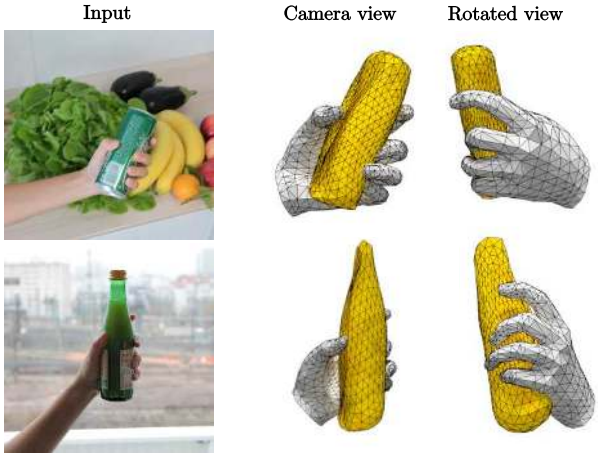

Figure 1: Our method jointly reconstructs hand and object meshes from a monocular RGB image. Note that the model generating the predictions for the above images, which we captured with an ordinary camera, was trained only on images from our synthetic dataset, ObMan.

Interactions impose constraints on relative configurations of hands and objects. For example, stable object grasps require contacts between hand and object surfaces, while solid objects prohibit penetration. In this work we exploit constraints imposed by object manipulations to reconstruct hands and objects as well as to model their interactions. We build on a parametric hand model, MANO [50], derived from 3D scans of human hands, that provides anthropomorphically valid hand meshes. We then propose

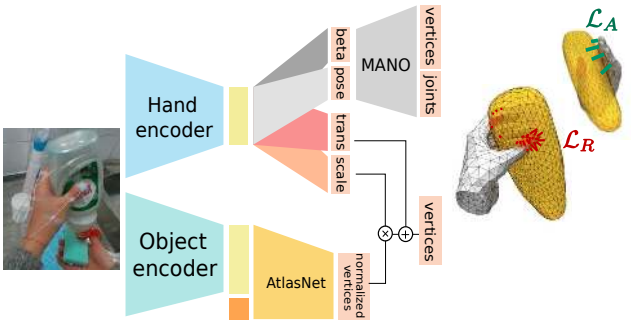

Figure 2: Our model predicts the hand and object meshes in a single forward pass in an end-to-end framework. The repulsion loss $\mathcal{L}_R$ penalizes interpenetration while the attraction loss $\mathcal{L}_A$ encourages the contact regions to be in contact with the object.

a differentiable MANO network layer enabling end-to-end learning of hand shape estimation. Equipped with the differentiable shape-based hand model, we next design a network architecture for joint estimation of hand shapes, object shapes and their relative scale and translation. We also propose a novel contact loss that penalizes penetrations and encourages contact between hands and manipulated objects. An overview of our method is illustrated in Figure 2.

Real images with ground truth shape for interacting hands and objects are difficult to obtain in practice. Existing datasets with hand-object interactions are either too small for training deep neural networks [64] or provide only partial 3D hand or object annotations [57]. The recent dataset by Garcia-Hernando *et al*. [8] provides 3D hand joints and meshes of 4 objects during hand-object interactions.

Synthetic datasets are an attractive alternative given their scale and readily-available ground truth. Datasets with synthesized hands have been recently introduced [28, 32, 73] but they do not contain hand-object interactions. We generate a new large-scale synthetic dataset with objects manipulated by hands: ObMan (*Ob*ject *Man*ipulation). We achieve diversity by automatically generating hand grasp poses for $2.7K$ everyday object models from 8 object categories. We adapt MANO to be able to interface it with an automatic grasp generation tool based on the GraspIt software [30]. ObMan is sufficiently large and diverse to support training and ablation studies of our deep models, and sufficiently realistic to generalize to real images. See Figure 1 for reconstructions obtained for real images when training our model on ObMan.

In summary we make the following contributions. First, we design the first end-to-end learnable model for joint 3D reconstruction of hands and objects from RGB data. Second, we propose a novel contact loss penalizing penetrations and encouraging contact between hands and objects. Third, we create a new large-scale synthetic dataset, ObMan, with hand-object manipulations. The ObMan dataset and our pre-trained models and code are publicly available[1].

[1] http://www.di.ens.fr/willow/research/obman/

2. Related work

In the following, we review methods that address hand and object reconstructions in isolation. We then present related works that jointly reconstruct hand-object interactions.

Hand pose estimation. Hand pose estimation has attracted a lot of research interest since the 90s [15, 44]. The availability of commodity RGB-D sensors [21, 43, 54] led to significant progress in estimating 3D hand pose given depth or RGB-D input [13, 20, 34, 35]. Recently, the community has shifted its focus to RGB-based methods [16, 32, 40, 55, 73]. To overcome the lack of 3D annotated data, many methods employed synthetic training images [7, 28, 32, 33, 73]. Similar to these approaches, we make use of synthetic renderings, but we additionally integrate object interactions.

3D hand pose estimation has often been treated as predicting 3D positions of *sparse* joints [16, 32, 73]. Unlike methods that predict only skeletons, our focus is to output a *dense* hand mesh to be able to infer interactions with objects. Very recently, Panteleris *et al*. [40] and Malik *et al*. [28] produce full hand meshes. However, [40] achieves this as a post-processing step by fitting to 2D predictions. Our hand estimation component is most similar to [28]. In contrast to [28], our method takes not depth but RGB images as input, which is more challenging and more general.

Regarding hand pose estimation in the presence of objects, Mueller *et al*. [32, 33] grasp 7 objects in a merged reality environment to render synthetic hand pose datasets. However, objects only serve the role of occluders, and the approach is difficult to scale to more object instances.

Object reconstruction. How to represent 3D objects in a CNN framework is an active research area. Voxels [29, 70], point clouds [59], and mesh surfaces [11, 19, 67] have been explored. We employ the latter since meshes allow better modeling of the interaction with the hand. AtlasNet [11] inputs vertex coordinates concatenated with image features and outputs a deformed mesh. More recently, Pixel2Mesh [67] explores regularizations to improve the perceptual quality of predicted meshes. Previous works mostly focus on producing accurate shape and they output the object in a normalized coordinate frame in a category-specific canonical pose. We employ a view-centered variant of [11] to handle generic object categories, without any category-specific knowledge. Unlike existing methods that typically input simple renderings of CAD models, such as ShapeNet [4], we work with complex images in the presence of hand occlusions. In-hand scanning [39, 51, 65, 69], while performed in the context of manipulation, focuses on object reconstruction and requires RGB-D video inputs.

Hand-object reconstruction. Joint reconstruction of hands and objects has been studied with multi-view RGB [2, 37, 68] and RGB-D input with either optimization [12, 13, 38, 42, 57, 63–65] or classification [46–49] approaches. These works use rigid objects, except for a few that use articulated [64] or deformable objects [63]. Focusing on contact

points, most works employ proximity metrics [57, 63, 64], while [46] directly regresses them from images, and [42] uses contact measurements on instrumented objects. [64] integrates physical constraints for penetration and contact, attracting fingers onto the object uni-directionally. On the contrary, [63] symmetrically attracts the fingertips and the object surface. The last two approaches evaluate all possible configurations of contact points and select the one that provides the most stable grasp [64] or best matches visual evidence [63]. Most related to our work, given an RGB image, Romero *et al*. [49] query a large synthetic dataset of rendered hands interacting with objects to retrieve configurations that match the visual evidence. Their method's accuracy, however, is limited by the variety of configurations contained in the database. In parallel work to ours [61] jointly estimates hand skeletons and 6DOF for objects. Our work differs from previous hand-object reconstruction methods mainly by incorporating an end-to-end learnable CNN architecture that benefits from a differentiable hand model and differentiable physical constraints on penetration and contact.

3. Hand-object reconstruction

As illustrated in Figure 2, we design a neural network architecture that reconstructs the hand-object configuration in a single forward pass from a rough image crop of a left hand holding an object. Our network architecture is split into two branches. The first branch reconstructs the object shape in a normalized coordinate space. The second branch predicts the hand mesh as well as the information necessary to transfer the object to the hand-relative coordinate system. Each branch has a ResNet18 [14] encoder pre-trained on ImageNet [52]. At test time, our model can process 20fps on a Titan X GPU. In the following, we detail the three components of our method: hand mesh estimation in Section 3.1, object mesh estimation in Section 3.2, and the contact between the two meshes in Section 3.3.

3.1. Differentiable hand model

Following the methods that integrate the SMPL parametric body model [25] as a network layer [17, 41], we integrate the MANO hand model [50] as a differentiable layer. MANO is a statistical model that maps pose (θ) and shape (β) parameters to a mesh. While the pose parameters capture the angles between hand joints, the shape parameters control the person-specific deformations of the hand; see [50] for more details.

Hand pose lives in a low-dimensional subspace [23, 50]. Instead of predicting the full 45-dimensional pose space, we predict 30 pose PCA components. We found that performance saturates at 30 PCA components and keep this value for all our experiments (see Appendix A.2).

Supervision on vertex and joint positions ($\mathcal{L}_{V_{Hand}}, \mathcal{L}_J$). The hand encoder produces an encoding Φ_{Hand} from an image. Given Φ_{Hand}, a fully connected network regresses θ and β. We integrate the mesh generation as a differentiable network layer that takes θ and β as inputs and outputs the hand vertices V_{Hand} and 16 hand joints. In addition to MANO joints, we select 5 vertices on the mesh as fingertips to obtain 21 hand keypoints J. We define the supervision on the vertex positions ($\mathcal{L}_{V_{Hand}}$) and joint positions ($\mathcal{L}_J$) to enable training on datasets where a ground truth hand surface is not available. Both losses are defined as the L2 distance to the ground truth. We use root-relative 3D positions as supervision for $\mathcal{L}_{V_{Hand}}$ and $\mathcal{L}_J$. Unless otherwise specified, we use the wrist defined by MANO as the root joint.

Regularization on hand shape ($\mathcal{L}_\beta$). Sparse supervision can cause extreme mesh deformations when the hand shape is unconstrained. We therefore use a regularizer, $\mathcal{L}_\beta = \|\beta\|^2$, on the hand shape to constrain it to be close to the average shape in the MANO training set, which corresponds to $\beta = \vec{0} \in \mathbb{R}^{10}$.

The resulting hand reconstruction loss $\mathcal{L}_{Hand}$ is the summation of all $\mathcal{L}_{V_{Hand}}$, $\mathcal{L}_J$ and $\mathcal{L}_\beta$ terms:

$$\mathcal{L}_{Hand} = \mathcal{L}_{V_{Hand}} + \mathcal{L}_J + \mathcal{L}_\beta. \quad (1)$$

Our experiments indicate benefits for all three terms (see Appendix A.1). Our hand branch also matches state-of-the-art performance on a standard benchmark for 3D hand pose estimation (see Appendix A.3).

3.2. Object mesh estimation

Following recent methods [19, 67], we focus on genus 0 topologies. We use AtlasNet [11] as the object prediction component of our neural network architecture. AtlasNet takes as input the concatenation of point coordinates sampled either on a set of square patches or on a sphere, and image features Φ_{Obj}. It uses a fully connected network to output new coordinates on the surface of the reconstructed object. AtlasNet explores two sampling strategies: sampling points from a sphere and sampling points from a set of squares. Preliminary experiments showed better generalization to unseen classes when input points were sampled on a sphere. In all our experiments we deform an icosphere of subdivision level 3 which has 642 vertices. AtlasNet was initially designed to reconstruct meshes in a canonical view. In our model, meshes are reconstructed in view-centered coordinates. We experimentally verified that AtlasNet can accurately reconstruct meshes in this setting (see Appendix B.1). Following AtlasNet, the supervision for object vertices is defined by the symmetric Chamfer loss between the predicted vertices and points randomly sampled on the ground truth external surface of the object.

Regularization on object shape ($\mathcal{L}_E, \mathcal{L}_L$). In order to reason about the inside and outside of the object, it is important to predict meshes with well-defined surfaces and good quality triangulations. However AtlasNet does not explicitly enforce constraints on mesh quality. We find that when

learning to model a limited number of object shapes, the triangulation quality is preserved. However, when training on the larger variety of objects of ObMan, we find additional regularization on the object meshes beneficial. Following [10, 18, 67] we employ two losses that penalize irregular meshes. We penalize edges with lengths different from the average edge length with an edge-regularization loss, $\mathcal{L}_E$. We further introduce a curvature-regularizing loss, $\mathcal{L}_L$, based on [18], which encourages the curvature of the predicted mesh to be similar to the curvature of a sphere (see details in Appendix B.2). We balance the weights of $\mathcal{L}_E$ and $\mathcal{L}_L$ by weights μ_E and μ_L respectively, which we empirically set to 2 and 0.1. These two losses together improve the quality of the predicted meshes, as we show in Figure A.4 of the appendix. Additionally, when training on the ObMan dataset, we first train the network to predict normalized objects, and then freeze the object encoder and the AtlasNet decoder while training the hand-relative part of the network. When training the objects in normalized coordinates, noted with n, the total object loss is:

$$\mathcal{L}^n_{Object} = \mathcal{L}^n_{V_{Obj}} + \mu_L \mathcal{L}_L + \mu_E \mathcal{L}_E. \quad (2)$$

Hand-relative coordinate system ($\mathcal{L}_S, \mathcal{L}_T$). Following AtlasNet [11], we first predict the object in a normalized scale by offsetting and scaling the ground truth vertices so that the object is inscribed in a sphere of fixed radius. However, as we focus on hand-object interactions, we need to estimate the object position and scale relative to the hand. We therefore predict translation and scale in two branches, which output the three offset coordinates for the translation (i.e., x, y, z) and a scalar for the object scale. We define $\mathcal{L}_T = \|T - \hat{T}\|_2^2$ and $\mathcal{L}_S = \|S - \hat{S}\|_2^2$, where $\hat{T}$ and $\hat{S}$ are the predicted translation and scale. T is the ground truth object centroid in hand-relative coordinates and S is the ground truth maximum radius of the centroid-centered object.

Supervision on object vertex positions ($\mathcal{L}^n_{V_{Obj}}, \mathcal{L}_{V_{Obj}}$). We multiply the AtlasNet decoded vertices by the predicted scale and offset them according to the predicted translation to obtain the final object reconstruction. Chamfer loss ($\mathcal{L}_{V_{Obj}}$) is applied after translation and scale are applied. When training in hand-relative coordinates the loss becomes:

$$\mathcal{L}_{Object} = \mathcal{L}_T + \mathcal{L}_S + \mathcal{L}_{V_{Obj}}. \quad (3)$$

3.3. Contact loss

So far, the prediction of hands and objects does not leverage the constraints that guide objects interacting in the physical world. Specifically, it does not account for our prior knowledge that objects can not interpenetrate each other and that, when grasping objects, contacts occur at the surface between the object and the hand. We formulate these contact constraints as a differentiable loss, $\mathcal{L}_{Contact}$, which can be directly used in the end-to-end learning framework. We incorporate this additional loss using a weight parameter μ_C, which we set empirically to 10.

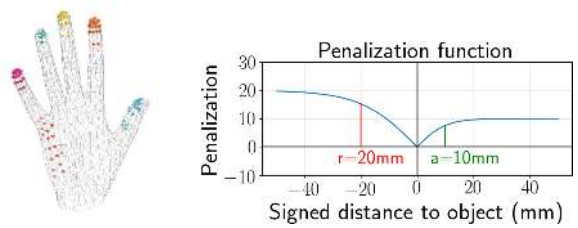

Figure 3: Left: Estimated contact regions from ObMan. We find that points that are often involved in contacts can be clustered into 6 regions on the palmar surface of the hand. Right: Generic shape of the penalization function emphasizing the role of the characteristic distances.

We rely on the following definition of distances between points. $d(v, V_{Obj}) = \inf_{w \in V_{Obj}} \|v - w\|_2$ denotes distances from point to set and $d(C, V_{Obj}) = \inf_{v \in C} d(v, V_{Obj})$ denotes distances from set to set. Moreover, we define a common penalization function $l_\alpha(x) = \alpha \tanh\left(\frac{x}{\alpha}\right)$, where α is a characteristic distance of action.

Repulsion ($\mathcal{L}_R$). We define a repulsion loss ($\mathcal{L}_R$) that penalizes hand and object *interpenetration*. To detect interpenetration, we first detect hand vertices that are inside the object. Since the object is a deformed sphere, it is watertight. We therefore cast a ray from the hand vertex and count the number of times it intersects the object mesh to determine whether it is inside or outside the predicted mesh [31]. $\mathcal{L}_R$ affects all hand vertices that belong to the interior of the object, which we denote $\mathrm{Int}(Obj)$. The repulsion loss is defined as:

$$\mathcal{L}_R(V_{Obj}, V_{Hand}) = \sum_{v \in V_{Hand}} \mathbb{1}_{v \in \mathrm{Int}(V_{Obj})} l_r(d(v, V_{Obj})),$$

where r is the repulsion characteristic distance, which we empirically set to $2\mathrm{cm}$ in all experiments.

Attraction ($\mathcal{L}_A$). We further define an attraction loss ($\mathcal{L}_A$) to penalize cases in which hand vertices are in the vicinity of the object but the surfaces are *not* in contact. This loss is applied only to vertices which belong to the exterior of the object $\mathrm{Ext}(Obj)$.

We compute statistics on the automatically-generated grasps described in the next section to determine which vertices on the hand are frequently involved in contacts. We compute for each MANO vertex how often across the dataset it is in the immediate vicinity of the object (defined as less than 3mm away from the object's surface). We find that by identifying the vertices that are close to the objects in at least 8% of the grasps, we obtain 6 regions of connected vertices $\{C_i\}_{i \in [\![1,6]\!]}$ on the hand which match the 5 fingertips and part of the palm of the hand, as illustrated in Figure 3 (left). The attraction term $\mathcal{L}_A$ penalizes distances from each of the regions to the object, allowing for sparse guidance towards the object's surface:

$$\mathcal{L}_A(V_{Obj}, V_{Hand}) = \sum_{i=1}^{6} l_a(d(C_i \cap \mathrm{Ext}(Obj), V_{Obj})). \quad (4)$$

We set a to 1cm in all experiments. For regions that are further from the hand than a threshold a, the attraction will significantly decrease and become negligible as the distance to the object further increases, see Figure 3 (right).

Our final contact loss $\mathcal{L}_{Contact}$ is a weighted sum of the attraction $\mathcal{L}_A$ and the repulsion $\mathcal{L}_R$ terms:

$$\mathcal{L}_{Contact} = \lambda_R \mathcal{L}_R + (1 - \lambda_R)\mathcal{L}_A, \tag{5}$$

where $\lambda_R \in [0, 1]$ is the contact weighting coefficient, e.g., $\lambda_R = 1$ means only the repulsion term is active. We show in our experiments that the balancing between attraction and repulsion is very important for physical quality.

Our network is first trained with $\mathcal{L}_{Hand} + \mathcal{L}_{Object}$. We then continue training with $\mathcal{L}_{Hand} + \mathcal{L}_{Object} + \mu_C \mathcal{L}_{Contact}$ to improve the physical quality of the hand-object interaction. Appendix C.1 gives further implementation details.

4. ObMan dataset

To overcome the lack of adequate training data for our models, we generate a large-scale synthetic image dataset of hands grasping objects which we call the *ObMan* dataset. Here, we describe how we scale automatic generation of hand-object images.

Objects. In order to find a variety of high-quality meshes of frequently manipulated everyday objects, we selected models from the ShapeNet [4] dataset. We selected 8 object categories of everyday objects (bottles, bowls, cans, jars, knifes, cellphones, cameras and remote controls). This results in a total of 2772 meshes which are split among the training, validation and test sets.

Grasps. In order to generate plausible grasps, we use the GraspIt software [30] following the methods used to collect the Grasp Database [9]. In the robotics community, this dataset has remained valuable over many years [53] and is still a reference for the fast synthesis of grasps given known object models [22, 27].

We favor simplicity and robustness of the grasp generation over the accuracy of the underlying model. The software expects a rigid articulated model of the hand. We transform MANO by separating it into 16 rigid parts, 3 parts for the phalanges of each finger, and one for the hand palm. Given an object mesh, GraspIt produces different grasps from various initializations. Following [9], our generated grasps optimize for the grasp metric but do not necessarily reflect the statistical distribution of human grasps. We sort the obtained grasps according to a heuristic measure (see Appendix C.2) and keep the two best candidates for each object. We generate a total of $21K$ grasps.

Body pose. For realism, we render the hand and the full body (see Figure 4). The pose of the hand is transferred to hands of the SMPL+H [50] model which integrates MANO to the SMPL [25, 50] statistical body model, allowing us to render realistic images of embodied hands. Although we zoom our cameras to focus on the hands, we vary the body poses to provide natural occlusions and coherent backgrounds. Body poses and shapes are varied by sampling from the same distribution as in SURREAL [66]; i.e., sampling poses from the CMU MoCap database [1] and shapes from CAESAR [45]. In order to maximize the viewpoint variability, a global rotation uniformly sampled in $SO(3)$ is also applied to the body. We translate the hand root joint to the camera's optical axis. The distance to the camera is sampled uniformly between 50 and 80cm.

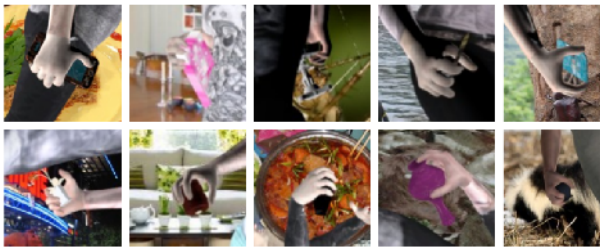

Figure 4: **ObMan:** large-scale synthetic dataset of hand-object interactions. We pose the MANO hand model [50] to grasp [30] a given object mesh. The scenes are rendered with variation in texture, lighting, and background.

Textures. Object textures are randomly sampled from the texture maps provided with ShapeNet [4] models. The body textures are obtained from the full body scans used in SURREAL [66]. Most of the scans have missing color values in the hand region. We therefore combine the body textures with 176 high resolution textures obtained from hand scans from 20 subjects. The hand textures are split so that textures from 14 subjects are used for training and 3 for test and validation sets. For each body texture, the skin tone of the hand is matched to the subject's face color. Based on the face skin color, we query in the HSV color space the 3 closest hand texture matches. We further shift the HSV channels of the hand to better match the person's skin tone.

Rendering. Background images are sampled from both the LSUN [72] and ImageNet [52] datasets. We render the images using Blender [3]. In order to ensure the hand and objects are visible we discard configurations if less than 100 pixels of the hand or if less than 40% of the object is visible.

For each hand-object configuration, we render object-only, hand-only, and hand-object images, as well as the corresponding segmentation and depth maps.

5. Experiments

We first define the evaluation metrics and the datasets (Sections 5.1, 5.2) for our experiments. We then analyze the effects of occlusions (Section 5.3) and the contact loss (Section 5.4). Finally, we present our transfer learning experiments from synthetic to real domain (Sections 5.5, 5.6).

5.1. Evaluation metrics

Our output is structured, and a single metric does not fully capture performance. We therefore rely on multiple

evaluation metrics.

Hand error. For hand reconstruction, we compute the mean end-point error (mm) over 21 joints following [73].

Object error. Following AtlasNet [11], we measure the accuracy of object reconstruction by computing the symmetric Chamfer distance (mm) between points sampled on the ground truth mesh and vertices of the predicted mesh.

Contact. To measure the physical quality of our joint reconstruction, we use the following metrics.

Penetration depth (mm), Intersection volume (cm^3): Hands and objects should not share the same physical space. To measure whether this rule is violated, we report the intersection volume between the object and the hand as well as the penetration depth. To measure the intersection volume of the hand and object we voxelize the hand and object using a voxel size of 0.5cm. If the hand and the object collide, the penetration depth is the maximum of the distances from hand mesh vertices to the object's surface. In the absence of collision, the penetration depth is 0.

Simulation displacement (mm): Following [64], we use physics simulation to evaluate the quality of the produced grasps. This metric measures the average displacement of the object's center of mass in a simulated environment [5] assuming the hand is fixed and the object is subjected to gravity. Details on the setup and the parameters used for the simulation can be found in [64]. Good grasps should be stable in simulation. However, stable simulated grasps can also occur if the forces resulting from the collisions balance each other. For estimating grasp quality, simulated displacement must be analyzed in conjunction with a measure of collision. If both displacement in simulation and penetration depth are decreasing, there is strong evidence that the physical quality of the grasp is improving (see Section 5.4 for an analysis). The reported metrics are averaged across the dataset.

5.2. Datasets

We present the datasets we use to evaluate our models. Statistics for each dataset are summarized in Table 1.

First-person hand benchmark (FHB). This dataset [8] is a recent video collection providing 3D hand annotations for a wide range of hand-object interactions. The joints are automatically annotated using magnetic sensors strapped on the hands, and which are visible on the RGB images. 3D mesh annotations are provided for four objects: three different bottles and a salt box. In order to ensure that the object being interacted with is unambiguously defined, we filter frames in which the manipulating hand is further than 1cm away from the manipulated object. We refer to this filtered dataset as FHB. As the milk bottle is a genus-1 object and is often grasped by its handle, we exclude this object from the experiments we conduct on contacts. We call this subset FHB_C. We use the same subject split as [8], therefore, each object is present in both the training and test splits.

The object annotations for this dataset suffer from some imprecisions. To investigate the range of the object ground truth error, we measure the penetration depth of the hand skeleton in the object for each hand-object configuration. We find that on the training split of FHB, the average penetration depth is 11.0mm (std=8.9mm). While we still report quantitative results on objects for completeness, the ground truth errors prevent us from drawing strong conclusions from reconstruction metric fluctuations on this dataset.

Hands in action dataset (HIC). We use a subset of the HIC dataset [64] which has sequences of a single hand interacting with objects. This gives us 4 sequences featuring manipulation of a sphere and a cube. We select the frames in which the hand is less than 5mm away from the object. We split this dataset into 2 training and 2 test sequences with each object appearing in both splits and restrict our predictions to the frames in which the minimal distance between hand and object vertices is below 5mm. For this dataset the hand and object meshes are provided. We fit MANO to the provided hand mesh, allowing for dense point supervision on both hands and objects.

	ObMan	FHB	FHB_C	HIC
#frames	$141K/6K$	$8420/9103$	$5077/5657$	$251/307$
#video sequences	-	$115/127$	$76/88$	$2/2$
#object instances	$1947/411$	4	3	2
real	no	yes	yes	yes

Table 1: Dataset details for train/test splits.

Training	Evaluation images H-img	HO-img	Training	Evaluation images O-img	HO-img
H-img ($\mathcal{L}_H$)	**10.3**	14.1	O-img ($\mathcal{L}_O$)	**0.0242**	0.0722
HO-img ($\mathcal{L}_H$)	11.7	**11.6**	HO-img ($\mathcal{L}_O$)	0.0319	**0.0302**

Table 2: We first show that training with occlusions is important when targeting images of hand-object interactions.

5.3. Effect of occlusions

For each sample in our synthetic dataset, in addition to the hand-object image (HO-img) we render two images of the corresponding isolated and unoccluded hand (H-img) or object (O-img). With this setup, we can systematically study the effect of occlusions on ObMan, which would be impractical outside of a synthetic setup.

We study the effect of objects occluding hands by training two networks, one trained on hand-only images and one on hand-object images. We report performance on both unoccluded and occluded images. A symmetric setup is applied to study the effect of hand occlusions on objects. Since the hand-relative coordinates are not applicable to experiments with object-only images, we study the normalized shape reconstruction, centered on the object centroid, and scaled to be inscribed in a sphere of radius 1.

Unsurprisingly, the best performance is obtained when both training and testing on unoccluded images as shown in Table 2. When both training and testing on occluded im-

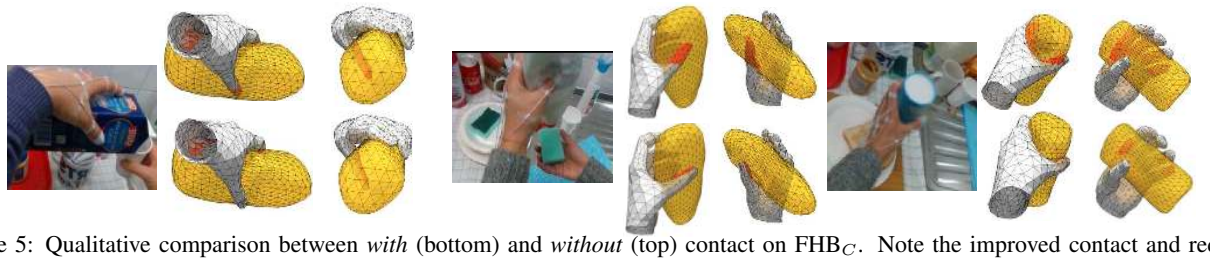

Figure 5: Qualitative comparison between *with* (bottom) and *without* (top) contact on FHB_C. Note the improved contact and reduced penetration, highlighted with red regions, with our contact loss.

	ObMan Dataset					FHB_C Dataset				
	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection Volume	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection Volume
No contact loss	11.6	641.5	9.5	31.3	12.3	28.1 ± 0.5	1579.2 ± 66.2	18.7 ± 0.6	51.2 ± 1.7	26.9 ± 0.2
Only attraction ($\lambda_R = 0$)	11.9	637.8	11.8	26.8	17.4	28.4 ± 0.6	1586.9 ± 58.3	22.7 ± 0.7	48.5 ± 3.2	41.2 ± 0.3
Only repulsion ($\lambda_R = 1$)	12.0	639.0	6.4	38.1	8.1	28.6 ± 0.8	1603.7 ± 49.9	6.0 ± 0.3	53.9 ± 2.3	7.1 ± 0.1
Attraction + Repulsion ($\lambda_R = 0.5$)	11.6	637.9	9.2	30.9	12.2	28.8 ± 0.8	1565.0 ± 65.9	12.1 ± 0.7	47.7 ± 2.5	17.6 ± 0.2

Table 3: We experiment with each term of the contact loss. Attraction ($\mathcal{L}_A$) encourages contacts between close points while repulsion ($\mathcal{L}_R$) penalizes interpenetration. λ_R is the repulsion weight, balancing the contribution of the two terms.

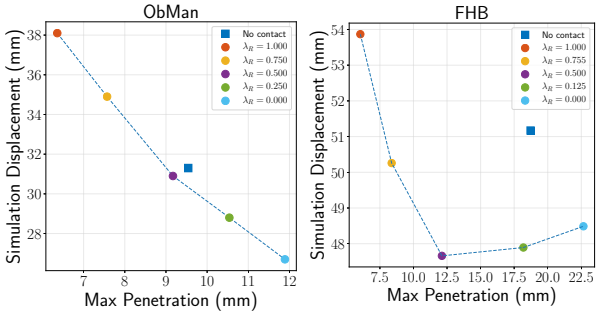

Figure 6: We examine the relative importance between the contact terms on the grasp quality metrics. Introducing a well-balanced contact loss improves upon the baseline on both max penetration and simulation displacement.

ages, reconstruction errors for hands and objects drop significantly, by 12% and 25% respectively. This validates the intuition that estimating hand pose and object shape in the presence of occlusions is a harder task.

We observe that for both hands and objects, the most challenging setting is training on unoccluded images while testing on images with occlusions. This shows that training with occlusions is crucial for accurate reconstruction of hands-object configurations.

5.4. Effect of contact loss

In the absence of explicit physical constraints, the predicted hands and objects have an average penetration depth of 9mm for ObMan and 19mm for FHB_C (see Table 3). The presence of interpenetration at test time shows that the model is not implicitly learning the physical rules governing hand-object manipulation. The differences in physical metrics between the two datasets can be attributed to the higher reconstruction accuracy for ObMan but also to the noisy object ground truth in FHB_C which produces penetrated and likely unstable 'ground truth' grasps.

In Figure 6, we study the effect of introducing our contact loss as a fine-tuning step. We linearly interpolate λ_R in $[\![0, 1]\!]$ to explore various relative weightings of the attraction and repulsion terms.

We find that using $\mathcal{L}_R$ in isolation efficiently minimizes the maximum penetration depth, reducing it by 33% for ObMan and 68% for FHB_C. This decrease occurs at the expense of the stability of the grasp in simulation. Symmetrically, $\mathcal{L}_A$ stabilizes the grasps in simulation, but produces more collisions between hands and objects. We find that equal weighting of both terms ($\mathcal{L}_R = 0.5$) improves *both* physical measures without negatively affecting the reconstruction metrics on both the synthetic and the real datasets, as is shown in Table 3 (last row). For FHB_C, for each metric we report the means and standard deviations for 10 random seeds.

We find that on the synthetic dataset, decreased penetration is systematically traded for simulation instability whereas for FHB_C increasing λ_R from 0 to 0.5 decreases depth penetration *without* affecting the simulation stability. Furthermore, for $\lambda_R = 0.5$, we observe significant qualitative improvements on FHB_c as seen in Figure 5.

5.5. Synthetic to real transfer

Large-scale synthetic data can be used to pre-train models in the absence of suitable real datasets. We investigate the advantages of pre-training on ObMan when targeting FHB and HIC. We investigate the effect of scarcity of real data on FHB by comparing pairs of networks trained using subsets of the real dataset. One is pre-trained on ObMan while the other is initialized randomly, with the exception of the encoders, which are pre-trained on ImageNet [52]. For these experiments, we do not add the contact loss and report means and standard deviations for 5 distinct random seeds. We find that pre-training on ObMan is beneficial in low data regimes, especially when less than 1000 images from the real dataset are used for fine-tuning, see Figure 8.

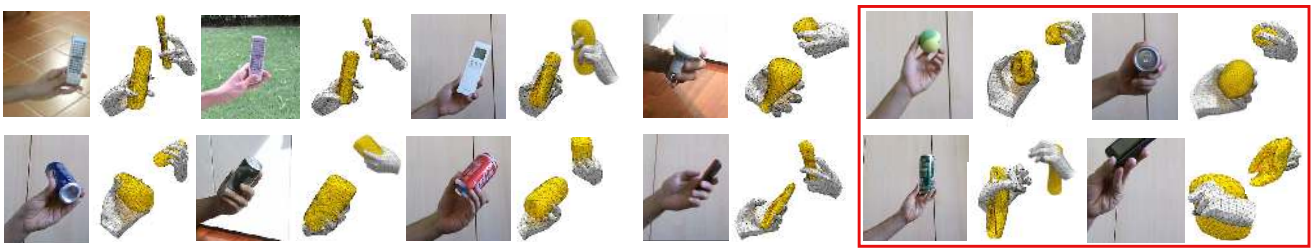

Figure 7: Qualitative results on CORe50. Our model, trained only on synthetic data, shows robustness to various hand poses, objects and scenes. Global hand pose and object outline are well estimated while fine details are missed. We present failure cases in the red box.

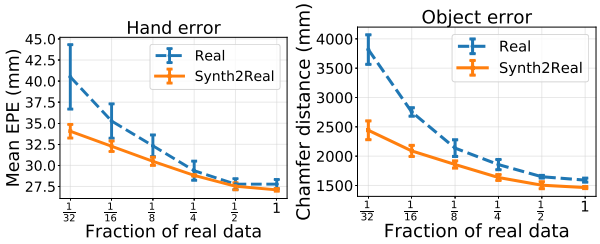

Figure 8: We compare training on FHB only (Real) and pre-training on synthetic, followed by fine-tuning on FHB (Synth2Real). As the amount of real data decreases, the benefit of pre-training increases. For both the object and the hand reconstruction, synthetic pre-training is critical in low-data regimes.

The HIC training set consists of only 250 images. We experiment with pre-training on variants of our synthetic dataset. In addition to ObMan, to which we refer as (a) in Figure 9, we render $20K$ images for two additional synthetic datasets, (b) and (c), which leverage information from the training split of HIC (d). We create (b) using our grasping tool to generate automatic grasps for each of the object models of HIC and (c) using the object and pose distributions from the training split of HIC. This allows to study the importance of sampling hand-object poses from the target distribution of the real data. We explore training on (a), (b), (c) with and without fine-tuning on HIC. We find that pre-training on all three datasets is beneficial for hand and object reconstructions. The best performance is obtained when pre-training on (c). In that setup, object performance outperforms training only on real images even *before* fine-tuning, and significantly improves upon the baseline after. Hand pose error saturates after the pre-training step, leaving no room for improvement using the real data. These results show that when training on synthetic data, similarity to the target real hand and pose distribution is critical.

5.6. Qualitative results on CORe50

FHB is a dataset with limited backgrounds, visible magnetic sensors and a very limited number of subjects and objects. In this section, we verify the ability of our model trained on ObMan to generalize to real data *without* fine-tuning. CORe50 [24] is a dataset which contains hand-object interactions with an emphasis on the variability of objects and backgrounds. However no 3D hand or object annotation is available. We therefore present qualitative results on this dataset. Figure 7 shows that our model generalizes across different object categories, including *lightbulb*, which does not belong to the categories our model was trained on. The global outline is well recovered in the camera view while larger mistakes occur in the perpendicular direction. More results can be found in Appendix D.

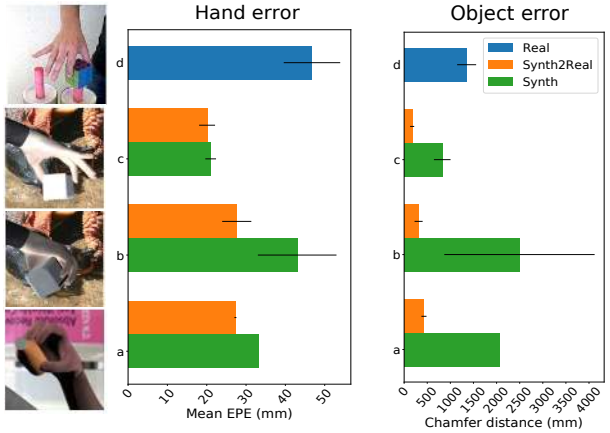

Figure 9: We compare the effect of training with and without fine-tuning on variants of our synthetic dataset on HIC. We illustrate each dataset (a, b, c, d) with an image sample, see text for definitions. Synthetic pre-training, whether or not the target distribution is matched, is always beneficial.

6. Conclusions

We presented an end-to-end approach for joint reconstruction of hands and objects given a single RGB image as input. We proposed a novel contact loss that enforces physical constraints on the interaction between the two meshes. Our results and the ObMan dataset open up new possibilities for research on modeling object manipulations. Future directions include learning grasping affordances from large-scale visual data, and recognizing complex and dynamic hand actions.

Acknowledgments. This work was supported in part by ERC grants ACTIVIA and ALLEGRO, the MSR-Inria joint lab, the Louis Vuitton ENS Chair on AI and the DGA project DRAAF. We thank Tsvetelina Alexiadis, Jorge Marquez and Senya Polikovsky from MPI for help with scan acquisition, Joachim Tesch for the hand-object rendering, Matthieu Aubry and Thibault Groueix for advices on AtlasNet, David Fouhey for feedback. MJB has received research gift funds from Intel, Nvidia, Adobe, Facebook, and Amazon. While MJB is a part-time employee of Amazon, his research was performed solely at, and funded solely by, MPI. MJB has financial interests in Amazon and Meshcapade GmbH.

References

[1] Carnegie-Mellon Mocap Database. http://mocap.cs.cmu.edu/. 5

[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. 2

[3] Blender Online Community. Blender - a 3D modelling and rendering package. http://www.blender.org. 5

[4] A. X. Chang, homas A. Funkhouser, eonidas J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*, 2015. 2, 5

[5] E. Coumans. Bullet real-time physics simulation, 2013. 6

[6] M. De La Gorce, D. J. Fleet, and N. Paragios. Model-based 3D hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1793–1805, 2011. 1

[7] E. Dibra, S. Melchior, T. Wolf, A. Balkis, A. C. Öztireli, and M. H. Gross. Monocular RGB hand pose inference from unsupervised refinable nets. In *CVPR Workshops*, 2018. 1, 2

[8] 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. 2, 6

[9] C. Goldfeder, M. T. Ciocarlie, H. Dang, and P. K. Allen. The Columbia grasp database. In *ICRA*, 2009. 5

[10] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. 3D-CODED : 3D correspondences by deep deformation. In *ECCV*, 2018. 4

[11] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *CVPR*, 2018. 2, 3, 4, 6

[12] H. Hamer, J. Gall, T. Weise, and L. Van Gool. An object-dependent hand pose prior from sparse training data. In *CVPR*, 2010. 2

[13] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *ICCV*, 2009. 1, 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 3

[15] T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 1996. 1, 2

[16] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018. 1, 2

[17] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 3

[18] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 4

[19] H. Kato, Y. Ushiku, and T. Harada. Neural 3D mesh renderer. In *CVPR*, 2018. 2, 3

[20] C. Keskin, F. Kıraç, Y. Kara, and L. Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV*, 2012. 1, 2

[21] Kinect. https://en.wikipedia.org/wiki/Kinect. 2

[22] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. In *The International Journal of Robotics Research*, 2015. 5

[23] J. Lin, Y. Wu, and T. S. Huang. Modeling the constraints of human hand motion. In *Proceedings of the Workshop on Human Motion (HUMO'00)*, 2000. 3

[24] V. Lomonaco and D. Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*, Proceedings of Machine Learning Research, 2017. 8

[25] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 3, 5

[26] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV*, 2000. 1

[27] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *Robotics: Science and Systems*, 2017. 5

[28] J. Malik, A. Elhayek, F. Nunnari, K. Varanasi, K. Tamaddon, A. Héloir, and D. Stricker. DeepHPS: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth. In *3DV*, 2018. 1, 2

[29] D. Maturana and S. Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *IROS*, 2015. 2

[30] A. T. Miller and P. K. Allen. Graspit! A versatile simulator for robotic grasping. *Robotics Automation Magazine, IEEE*, 11:110 – 122, 2004. 2, 5

[31] T. Möller and B. Trumbore. Fast, minimum storage ray-triangle intersection. *J. Graph. Tools*, 1997. 4

[32] F. Mueller, 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

[33] 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. 2

[34] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *Proc. Computer Vision Winter Workshop*, 2015. 2

[35] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *ICCV*, 2015. 2

[36] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *BMVC*, 2011. 1

[37] 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. 2

[38] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012. 2

[39] P. Panteleris, N. Kyriazis, and A. A. Argyros. 3D tracking of human hands in interaction with unknown objects. In *BMVC*, 2015. 2

[40] 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. 1, 2

[41] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *CVPR*, 2018. 3

[42] T. Pham, N. Kyriazis, A. A. Argyros, and A. Kheddar. Hand-

object contact force estimation from markerless visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2, 3

[43] PrimeSense. https://en.wikipedia.org/wiki/PrimeSense. 2

[44] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *ECCV*, 1994. 1, 2

[45] K. M. Robinette, S. Blackwell, H. Daanen, M. Boehmer, S. Fleming, T. Brill, D. Hoeferlin, and D. Burnsides. Civilian American and European Surface Anthropometry Resource (CAESAR) final report. Technical Report AFRL-HE-WP-TR-2002-0169, US Air Force Research Laboratory, 2002. 5

[46] G. Rogez, J. S. S. III, and D. Ramanan. Understanding everyday hands in action from RGB-D images. In *ICCV*, 2015. 2, 3

[47] G. Rogez, M. Khademi, J. S. Supančič III, J. M. M. Montiel, and D. Ramanan. 3D hand pose detection in egocentric RGB-D images. In *ECCV Workshop on Consumer Depth Cameras for Computer Vision*, 2014.

[48] G. Rogez, J. S. Supančič III, and D. Ramanan. First-person pose recognition using egocentric workspaces. In *CVPR*, 2015.

[49] J. Romero, H. Kjellström, and D. Kragic. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *ICRA*, 2010. 2, 3

[50] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 1, 3, 5

[51] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. *ACM Transactions on Graphics (TOG)*, 21(3):438–446, 2002. 2

[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3, 5, 7

[53] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 2012. 5

[54] J. Shotton, A. Fitzgibbon, A. Blake, A. Kipman, M. Finocchio, B. Moore, and T. Sharp. Real-time human pose recognition in parts from a single depth image. In *CVPR*, 2011. 2

[55] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 1, 2

[56] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 1

[57] 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. 2, 3

[58] B. Stenger, P. R. Mendonça, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *CVPR*, 2001. 1

[59] H. Su, H. Fan, and L. Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017. 2

[60] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013. 1

[61] B. Tekin, F. Bogo, and M. Pollefeys. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *CVPR*, 2019. 3

[62] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169:1–169:10, 2014. 1

[63] A. Tsoli and A. Argyros. Joint 3D tracking of a deformable object in interaction with a hand. In *ECCV*, 2018. 2, 3

[64] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016. 2, 3, 6

[65] D. Tzionas and J. Gall. 3D object reconstruction from hand-object interactions. In *ICCV*, 2015. 2

[66] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *CVPR*, 2017. 5

[67] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV*, 2018. 2, 3, 4

[68] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)*, 32(4):43:1–43:14, 2013. 2

[69] T. Weise, T. Wismer, B. Leibe, and L. Van Gool. Online loop closure for real-time interactive 3D scanning. *Computer Vision and Image Understanding*, 115(5):635–648, 2011. 2

[70] J. Wu, Y. Wang, T. Xue, X. Sun, W. T. Freeman, and J. B. Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *NIPS*, 2017. 2

[71] Y. Wu, J. Y. Lin, and T. S. Huang. Capturing natural hand articulation. In *ICCV*, 2001. 1

[72] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv:1506.03365*, 2015. 5

[73] C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, 2017. 1, 2, 6