Real-time Hand Tracking under Occlusion from an Egocentric RGB-D Sensor

Franziska Mueller[1,2] Dushyant Mehta[1,2] Oleksandr Sotnychenko[1]
Srinath Sridhar[1] Dan Casas[3] Christian Theobalt[1]
[1]Max Planck Institute for Informatics [2]Saarland University [3]Universidad Rey Juan Carlos
{frmueller, dmehta, osotnych, ssridhar, dcasas, theobalt}@mpi-inf.mpg.de

Abstract

We present an approach for real-time, robust and accurate hand pose estimation from moving egocentric RGB-D cameras in cluttered real environments. Existing methods typically fail for hand-object interactions in cluttered scenes imaged from egocentric viewpoints—common for virtual or augmented reality applications. Our approach uses two subsequently applied Convolutional Neural Networks (CNNs) to localize the hand and regress 3D joint locations. Hand localization is achieved by using a CNN to estimate the 2D position of the hand center in the input, even in the presence of clutter and occlusions. The localized hand position, together with the corresponding input depth value, is used to generate a normalized cropped image that is fed into a second CNN to regress relative 3D hand joint locations in real time. For added accuracy, robustness and temporal stability, we refine the pose estimates using a kinematic pose tracking energy. To train the CNNs, we introduce a new photorealistic dataset that uses a merged reality approach to capture and synthesize large amounts of annotated data of natural hand interaction in cluttered scenes. Through quantitative and qualitative evaluation, we show that our method is robust to self-occlusion and occlusions by objects, particularly in moving egocentric perspectives.

1. Introduction

Estimating the full articulated 3D pose of hands is becoming increasingly important due to the central role that hands play in everyday human activities. Applications in activity recognition [21], motion control [42], human–computer interaction [25], and virtual/augmented reality (VR/AR) require real-time and accurate hand pose estimation under challenging conditions. Spurred by recent developments in commodity depth sensing, several methods that use a single RGB-D camera have been proposed [33, 26, 30, 17, 4, 37]. In particular, methods that use Convolutional Neural Networks (CNNs), possibly in combination with model-based hand tracking, have been shown to work well for static, third-person viewpoints in uncluttered scenes [34, 24, 13], *i.e.*, mostly for free hand motion in mid-air, a setting that is uncommon in natural hand interaction.

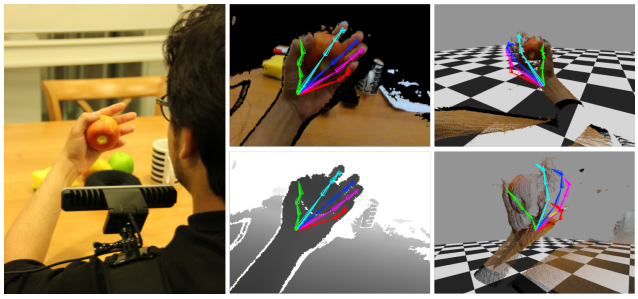


Figure 1: We present an approach to track the full 3D pose of the hand from egocentric viewpoints (left), a challenging problem due to additional self-occlusions, occlusions from objects and background clutter. Our method can reliably track the hand in 3D even under such conditions using only RGB-D input. Here we show tracking results overlaid with color and depth map (center), and visualized from virtual viewpoints (right).

However, real-time hand pose estimation from **moving**, **first-person** camera viewpoints in **cluttered real-world scenes** where the hand is often occluded as it naturally interacts with objects, remains an unsolved problem. We define first-person or **egocentric** viewpoints as those that would typically be imaged by cameras mounted on the head (for VR/AR applications), shoulder, or chest (see Figure 1). Occlusions, cluttered backgrounds, manipulated objects, and field-of-view limitations make this scenario particularly challenging. CNNs are a promising method to tackle this problem but typically require large amounts of *annotated* data which is hard to obtain since markerless hand tracking (even with multiple views), and manual annotation on a large scale is infeasible in egocentric settings due to (self-)occlusions, cost, and time. Even semi-

automatic annotation approaches [12] would fail if large parts of the hand are occluded. Photorealistic synthetic data, on the other hand, is inexpensive, easier to obtain, removes the need for manual annotation, and can produce accurate ground truth even under occlusion.

In this paper, we present, to our knowledge, the first method that supports **real-time** egocentric hand pose estimation in real scenes with cluttered backgrounds, occlusions, and complex hand-object interactions using a single commodity RGB-D camera. We divide the task of per-frame hand pose estimation into: (1) hand localization, and (2) 3D joint location regression. Hand localization, an important task in the presence of scene clutter, is achieved by a CNN that estimates the 2D image location of the center of the hand. Further processing results in an image-level bounding box around the hand and the 3D location of the hand center (or of the occluding object directly in front of the center). This output is fed into a second CNN that regresses the relative 3D locations of the 21 hand joints. Both CNNs are trained with large amounts of fully annotated data which we obtain by combining hand-object interactions with real cluttered backgrounds using a new **merged reality** approach. This increases the realism of the training data since users can perform motions to mimic manipulating a virtual object using live feedback of their hand pose. These motions are rendered from novel egocentric views using a framework that photorealistically models RGB-D data of hands in natural interaction with objects and clutter.

The 3D joint location predictions obtained from the CNN are accurate but suffer from kinematic inconsistencies and temporal jitter expected in single frame pose estimation methods. To overcome this, we refine the estimated 3D joint locations using a fast inverse kinematics pose tracking energy that uses 3D and 2D joint location constraints to estimate the joint angles of a temporally smooth skeleton. Together, this results in the first real-time approach for smooth and accurate hand tracking even in cluttered scenes and from moving egocentric viewpoints. We show the accuracy, robustness, and generality of our approach on a new benchmark dataset with moving egocentric cameras in real cluttered environments. In sum, our contributions are:

- A novel method that localizes the hand and estimates, in real time, the 3D joint locations from egocentric viewpoints, in clutter, and under strong occlusions using two CNNs. A kinematic pose tracking energy further refines the pose by estimating joint angles of a temporally smooth tracking.
- A photorealistic data generation framework for synthesizing large amounts of annotated RGB-D training data of hands in natural interaction with objects and clutter.
- Extensive evaluation on our new annotated real benchmark dataset *EgoDexter* featuring egocentric cluttered scenes and interaction with objects.

2. Related work

Hand pose estimation has a rich history due to its applications in human–computer interaction, motion control and activity recognition. However, most previous work estimates hand pose in mid-air and in uncluttered scenes with third-person viewpoints, making occlusions less of an issue. We first review the prior art for this simpler setting (*free hand tracking*) followed by a discussion of work in the harder hand-object and egocentric settings.

Free Hand Tracking: Many approaches for free hand tracking resort to multiple RGB cameras to overcome self-occlusions and achieve high accuracy [28, 1, 39]. However, single depth or RGB-D cameras are preferred since multiple cameras are cumbersome to setup and use. Methods that use generative pose tracking have been successful for free hand tracking with only an RGB-D stream [14, 17, 30, 32]. However, these approaches fail under typical fast motions, and occlusions due to objects and clutter. To overcome this, most recent approaches rely solely on learning-based methods or combine them with generative pose tracking. Random forests are a popular choice [9, 31, 29, 40, 38] due to their capacity but still result in kinematically inconsistent and jittery pose estimates. Many methods overcome this limitation through combination with a generative pose tracking strategy [26, 17, 33]. All of the above approaches fail to work under occlusions arising from objects and scene clutter. Recent deep learning methods promise large learning capacities for hand pose estimation [34, 4, 24, 43, 41, 13]. However, generating enough examples for supervised training remains a challenge. Commercial systems that claim to work for egocentric viewpoints [11] fail under large occlusions, see Section 6.

Hand Tracking under Challenging Conditions: Hand pose estimation under challenging scene, background, and camera conditions different from third-person mid-air tracking remains an unsolved problem. Some methods can track hands even when they interact with objects [5, 22], but they are limited to slow motions and limited articulation. A method for real-time joint tracking of hands and objects from third-person viewpoints was recently proposed [27], but is limited to known objects and small occlusions. Methods for capturing complex hand-object interactions and object scanning were proposed [15, 1, 10, 36, 35, 16]. However, these are offline methods and their performance in egocentric cluttered scenarios is unknown.

Using egocentric cameras for human performance capture has gained attention due to ready availability of consumer wearable cameras [18]. Sridhar *et al*. [26] showed a working example of real-time egocentric tracking in uncluttered scenes. Rogez *et al*. [19, 20] presented one of the first methods to achieve this in cluttered scenes with natural hand-object interactions pioneering the use of synthetic images for training a machine learning approach for diffi-

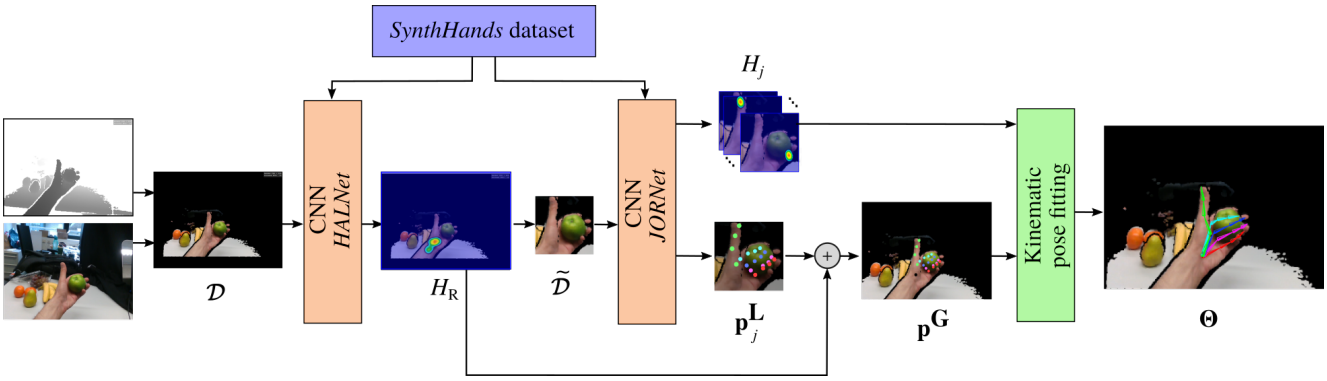


Figure 2: Overview: Starting from an RGB-D frame, we initially regress the 2D hand position heatmap using our CNN *HALNet* and compute a cropped frame. A second CNN, *JORNet*, is used to predict root-relative 3D joint positions as well as 2D joint heatmaps. Both CNNs are trainned with our new *SynthHands* dataset. Finally, we use a pose tracking step to obtain the joint angles of a kinematic skeleton.

cult egocentric views. However, this work was not meant for real-time tracking. We introduce an approach to leverage large amounts of synthetic training data to achieve *real-time*, temporally consistent hand tracking, even under challenging occlusion conditions.

3. Overview

Our goal is to estimate the full 3D articulated pose of the hand imaged with a single commodity RGB-D sensor. We use the RGB and depth channels from the Intel RealSense SR300 [7], both with a resolution of 640×480 pixels and captured at 30 Hz. We formulate hand pose estimation as an energy minimization problem that incorporates per-frame pose estimates into a temporal tracking framework. The goal is to find the joint angles of a kinematic hand skeleton (Section 3.1) that best represent the input observation. Similar strategies have been shown to be successful in state-of-the-art methods [33, 26, 27, 17] that use per-frame pose estimation to initialize a tracker that refines and regularizes the joint angles of a kinematic skeleton for free hand tracking. However, the per-frame pose estimation components of these methods struggle under strong occlusions, hand-object interactions, scene clutter, and moving egocentric cameras. We overcome this limitation by combining a CNN-based 3D pose regression framework, that is tailored for this challenging setting, with a kinematic skeleton tracking strategy for temporally stable results.

We divide the task of hand pose estimation into several subtasks (Figure 2). First, *hand localization* (Section 4.1) is achieved by a CNN that estimates an image-level heatmap (that encodes position probabilities) of the **root** — a point which is either the hand center (knuckle of the middle finger, shown with a star shape in Figure 3a) or a point on an object that occludes the hand center. The 2D and 3D root positions are used to extract a normalized cropped image of the hand. Second, *3D joint regression* (Section 4.2) achieved with a CNN that regresses root-relative 3D joint locations from the cropped hand image. Both CNNs are trained with large amounts of annotated data which were generated with a novel framework to automatically generate 3D hand joint motion with natural hand interaction (Section 4.4). Finally, the regressed 3D joint positions are used in a kinematic pose tracking framework (Section 5) to obtain temporally smooth tracking of the hand motion.

3.1. Hand Model

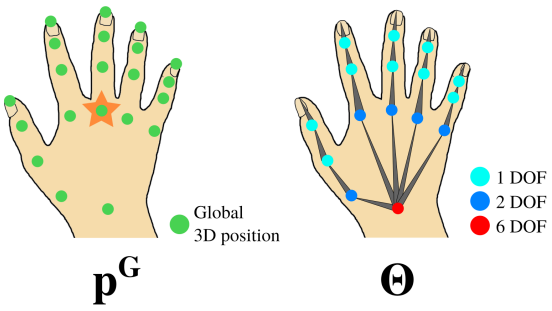
To ensure a consistent representation for both joint locations (predicted by the CNNs) and joint angles (optimized during tracking), we use a kinematic skeleton. As shown in Figure 3, we model the hand using a hierarchy of bones (gray lines) and joints (circles). The 3D joint locations are used as constraints in a kinematic pose tracking step that estimates temporally smooth joint angles of a kinematic skeleton. In our implementation, we use a kinematic skeleton with 26 degrees of freedom (DOF), which includes 6 for global translation and rotation, and 20 joint angles, stored in a vector $\boldsymbol{\Theta}$, as shown in Figure 3b. To fit users with different hand shapes and sizes, we perform a quick calibration step to fix the length of the bones for different users.

4. Single Frame 3D Pose Regression

The goal of 3D pose regression is to estimate the 3D joint locations of the hand at each frame of the RGB-D input. To achieve this, we first create a *colored depth map* $\mathcal{D}$, from the raw input produced by commodity RGB-D cameras (*e.g.*, Intel RealSense SR300). We define $\mathcal{D}$ as

$$\mathcal{D} = \texttt{colormap}(\mathbf{R}, \mathbf{G}, \mathbf{B}, \mathbf{Z}), \tag{1}$$

where $\texttt{colormap}(\cdot)$ is a function, that depends on the camera calibration parameters, to map each pixel in the color



(a) Global 3D positions (b) Kinematic skeleton

Figure 3: We use two different, but consistent, representations to model the hands. Our 3D joint regression step outputs $J = 21$ global 3D joint locations, shown in (a) in green, which are later used to estimate the joint angles of a kinematic skeleton hand model, shown in (b). The orange star depicts the joint used as a hand root.

image plane onto the depth map $\mathbf{Z}$. Computing $\mathcal{D}$ allows us to ignore camera-specific variations in extrinsic parameters. We also downsample $\mathcal{D}$ to a resolution of 320×240 to aid real-time performance. We next describe our pose regression approach that is robust even in challenging cluttered scenes with notable (self-)occlusions of the hand. As we show in the evaluation (Section 6), using a two step approach to first localize the hand in full-frame input and subsequently estimate 3D pose outperforms using a single CNN for both tasks.

4.1. Hand Localization

The goal of the first part of pose regression is to localize the hand in challenging cluttered input frames resulting in a bounding box around the hand and 3D root location. Given a colored depth map $\mathcal{D}$, we compute

$$\widetilde{\mathcal{D}} = \texttt{imcrop}(\mathcal{D}, H_{\text{R}}), \tag{2}$$

where H_{R} is a heatmap encoding the position probability of the 2D hand root and $\texttt{imcrop}(\cdot)$ is a function that crops the hand area of the input frame. In particular, we estimate H_{R} using a CNN which we call *HALNet* (HAnd Localization Net). The $\texttt{imcrop}(\cdot)$ function picks the image-level heatmap maximum location $\phi(H_{\text{R}}) = (u, v)$ and uses the associated depth z in $\mathcal{D}$ to compute a depth-dependent crop, the side length of which is inversely proportional to the depth and contains the hand. Additionally, $\texttt{imcrop}(\cdot)$ also normalizes the depth component of the cropped image by subtracting z from all pixels.

HALNet uses an architecture derived from *ResNet50* [6] which has been shown to have a good balance between accuracy and computational cost [2]. We reduced the number of residual blocks to 10 to achieve real-time framerate while maintaining high accuracy. We train this network using *SynthHands*, a new photorealistic dataset with ample variance across many dimensions such as hand pose, skin color, objects, hand-object interaction and shading details. See Sections 4.3 and 4.4, and the supplementary document for training and architecture details.

Post Processing: To make the root maximum location robust over time, we add an additional step to prevent outliers from affecting 3D joint location estimates. We maintain a history of maxima locations and label them as *confident* or *uncertain* based on the following criterion. If the likelihood value of the heatmap maximum at a frame t is < 0.1 and it occurs at > 30 pixels from the previous maximum then it is marked as uncertain. If a maximum location is uncertain, we update it as

$$\phi_t = \phi_{t-1} + \delta^k \frac{\phi_{c-1} - \phi_{c-2}}{||\phi_{c-1} - \phi_{c-2}||}, \tag{3}$$

where $\phi_t = \phi(H^t_{\text{R}})$ is the updated 2D maximum location at the frame t, ϕ_{c-1} is the last *confident* maximum location, k is the number of frames elapsed since the last confident maximum, and δ is a decay factor to progressively down-weight uncertain maxima. We empirically set $\delta = 0.98$ and use this value in all our results.

4.2. 3D Joint Regression

Starting from a cropped and normalized input $\widetilde{\mathcal{D}}$ that contains a hand, potentially partially occluded, our goal is to regress the global 3D hand joint position vector $\mathbf{p}^{\mathbf{G}} \in \mathbb{R}^{3\times J}$. We use a CNN, referred to as *JORNet* (JOint Regression Net), to predict per-joint 3D root-relative positions $\mathbf{p}^{\mathbf{L}} \in \mathbb{R}^{3\times J}$ in $\widetilde{\mathcal{D}}$. Additionally, *JORNet* also regresses per-joint 2D position likelihood heatmaps $\mathbf{H} = \{H_j\}_{j=1}^{J}$, which will be used to regularize the predicted 3D joint positions in a later step. We obtain global 3D joint positions $\mathbf{p}^{\mathbf{G}}_j = \mathbf{p}^{\mathbf{L}}_j + \mathbf{r}$, where $\mathbf{r}$ is the global position of the hand center (or a point on an occluder) obtained by backprojecting the 2.5D hand root position (u, v, z) to 3D. *JORNet* uses the same architecture as *HALNet* and is trained with the same data. See Sections 4.3 and 4.4 for training details, and the supplementary document for architecture details.

4.3. *SynthHands* Dataset

Supervised learning methods, including CNNs, require large amounts of training data in order to learn all the variation exhibited in real hand motion. Fully annotated real data would be ideal for this purpose but it is time consuming to manually annotate data and annotation quality may not always be good [12]. To circumvent this problem, existing methods [19, 20] have used synthetic data. Despite the advances made, existing datasets are constrained in a number of ways: they typically show unnatural mid-air motions, no

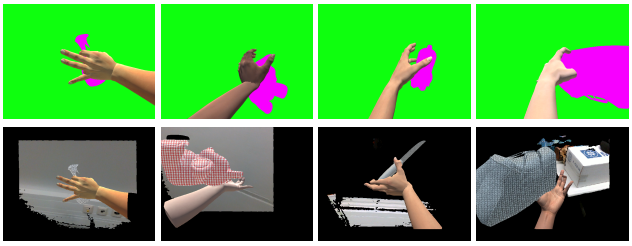


Figure 4: Our *SynthHands* dataset is created by posing a photorealistic hand model with real hand motion data. Virtual objects are incorporated into the 3D scenario. To allow data augmentation, we output object foreground and scene background appearance as a constant plain color (top row), which are composed with shading details and randomized textures in a postprocessing step (bottom row).

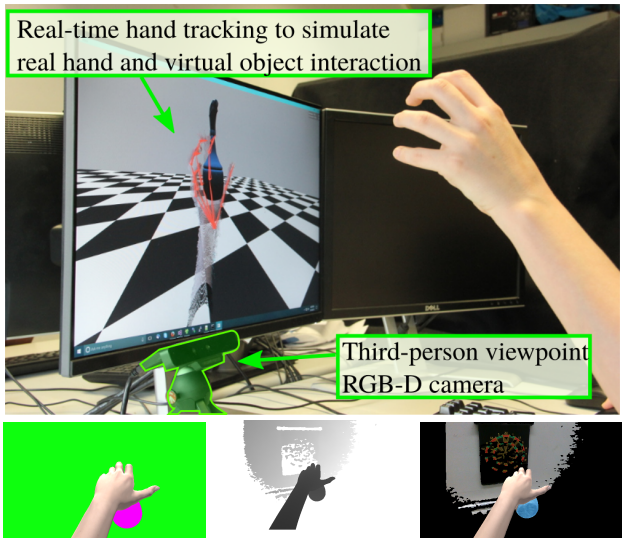


Figure 5: Our *SynthHands* dataset has accurate annotated data of a hand interacting with objects. We use a merged reality framework to track a real hand, where all joint positions are annotated, interacting with a virtual object (top). Synthetic images are rendered with chroma key-ready colors, enabling data augmentation by composing the rendered hand with varying object texture and real cluttered backgrounds (bottom).

complex hand-object interactions, and do not model realistic background clutter and noise.

We propose a new dataset, *SynthHands*, that combines real captured hand motion (retargeted to a virtual hand model) with natural backgrounds and virtual objects to sample all important dimensions of variability at previously unseen granularity. It captures the variations in natural hand motion such as pose, skin color, shape, texture, background clutter, camera viewpoint, and hand-object interactions. We now highlight some of the unique features of this dataset that make it ideal for supervised training of learning-based methods.

Natural Hand Motions: Instead of using static hand poses [20], we captured real, non-occluded, hand motion in mid-air from a third-person viewpoint, with a state-of-the-art real-time markerless tracker [26]. These motions were subsequently re-targeted onto a photorealistic synthetic hand rigged by an artist. Because we pose the synthetic hand using the captured hand motion, it mimics real hand motions and increases dataset realism.

Hand Shape and Color: Hand shape and skin color exhibit large variation across users. To simulate real world diversity, *SynthHands* contains skin textures randomly sampled from 12 different skin tones. We also sample variation in other anatomical features (*e.g.*, male hands are typically bigger and may contain more hair) in the data. Finally, we model hand shape variation by randomly applying a scaling parameter $\beta \in [0.8, 1.2]$ along each dimension of a default hand mesh.

Egocentric Viewpoint: Synthetic data has the unique advantage that we can render from arbitrary camera viewpoints. In order to support difficult egocentric views, we setup 5 virtual cameras that mimic different egocentric perspectives. The virtual cameras generate RGB-D images from this perspective while also simulating sensor noise and camera calibration parameters.

Hand-Object Interactions: We realistically simulate hand-object interactions by using a merged reality approach to track real hand motion interacting with virtual objects. We achieve this by leveraging the real-time capability of existing hand tracking solutions [26] to show the user's hand interacting with a virtual on-screen object. Users perform motions such as object grasping and manipulation, thus simulating real hand-object interactions (see Figure 5).

Object Shape and Appearance: *SynthHands* contains interactions with a total of 7 different virtual objects in various locations, rotations and scale configurations. To enable augmentation of the object appearance to increase dataset variance, we render the object albedo (*i.e.*, pink in Figure 4) and shading layers separately. We use chroma keying to replace the pink object albedo with a texture randomly sampled from a set of 145 textures and combining it with the shading image. Figure 4 shows some examples of the data before and after augmentation. Importantly, note that *SynthHands* does not contain 3D scans of the real test objects nor 3D models of similar objects used for evaluation in Section 6. This demonstrates that our approach generalizes to unseen objects.

Real Backgrounds: Finally, we simulate cluttered scenes and backgrounds by compositing the synthesized hand-object images with real RGB-D captures of real backgrounds, including everyday desktop scenarios, offices, corridors and kitchens. We use chroma keying to replace the

default background (green in Figure 4) with the captured backgrounds.

Our data generation framework is built using the Unity Game Engine and uses a rigged hand model distributed by Leap Motion [11]. In total, *SynthHands* contains roughly 220,000 RGB-D images exhibiting large variation seen in natural hands and interactions. Please see the supplementary document for more information and example images.

4.4. Training

Both *HALNet* and *JORNet* are trained on the *SynthHands* dataset using the Caffe framework [8], and the AdaDelta solver with a momentum of 0.9 and weight decay factor of 0.0005. The learning rate is tapered down from 0.05 to 0.000025 during the course of the training. For training *JORNet*, we used the ground truth (u, v) and z of the hand root to create the normalized crop input. To improve robustness, we also add uniform noise ($\in [-25, 25]$ mm) to the backprojected 3D root position in the *SynthHands* dataset. We trained *HALNet* for 45,000 iterations and *JORNet* for 60,000 iterations. The final networks were chosen as the ones with the lowest loss values. The layers in our networks that are similar to *ResNet50* are initialized with weights of the original *ResNet50* architecture trained on ImageNet [23]. For the other layers, we initialize the weights randomly. For details of the loss weights used and the taper scheme, please see the supplementary document.

5. Hand Pose Optimization

The estimated per-frame global 3D joint positions $\mathbf{p}^{\mathbf{G}}$ are not guaranteed to be temporally smooth nor do they have consistent inter-joint distances (*i.e.*, bone lengths) over time. We mitigate this by fitting a kinematic skeleton parameterized by joint angles $\mathbf{\Theta}$, shown in Figure 3b, to the regressed 3D joint positions. Additionally, we refine the fitting by leveraging the 2D heatmap output from *JORNet* as an additional contraint and regularize it using joint limit and smoothness constraints. In particular, we seek to minimize

$$\mathcal{E}(\mathbf{\Theta}) = E_{\text{data}}(\mathbf{\Theta}, \mathbf{p}^{\mathbf{G}}, \mathbf{H}) + E_{\text{reg}}(\mathbf{\Theta}), \tag{4}$$

where E_{data} is the data term that incorporates both the 3D positions and 2D heatmaps

$$E_{\text{data}}(\mathbf{\Theta}, \mathbf{p}^{\mathbf{G}}, \mathbf{H}) = w_{\text{p3}} E_{\text{pos3D}}(\mathbf{\Theta}, \mathbf{p}^{\mathbf{G}}) + w_{\text{p2}} E_{\text{pos2D}}(\mathbf{\Theta}, \mathbf{H}). \tag{5}$$

The first term E_{pos3D} minimizes the 3D distance between each predicted joint location $\mathbf{p}_j^{\mathbf{G}}$ and its corresponding position $\mathcal{M}(\mathbf{\Theta})_j$ in the kinematic skeleton set to pose $\mathbf{\Theta}$

$$E_{\text{pos3D}}(\mathbf{\Theta}) = \sum_{j=1}^{J} ||\mathcal{M}(\mathbf{\Theta})_j - \mathbf{p}_j^{\mathbf{G}}||_2^2. \tag{6}$$

The second data term, E_{pos2D}, minimizes the 2D distance between each joint heatmap maximum $\phi(H_j)$ and the projected 2D location of the corresponding joint in the kinematic skeleton

$$E_{\text{pos2D}}(\mathbf{\Theta}) = \sum_{j=1}^{J} ||\pi(\mathcal{M}(\mathbf{\Theta})_j) - \phi(H_j))||_2^2, \tag{7}$$

where π projects the joint onto the image plane. We empirically tuned the weights for the different terms as: $w_{p3} = 0.01$ and $w_{p2} = 5 \times 10^{-7}$.

We regularize the data terms by enforcing joint limits and temporal smoothness constraints

$$E_{\text{reg}}(\mathbf{\Theta}) = w_{\text{l}} E_{\text{lim}}(\mathbf{\Theta}) + w_{\text{t}} E_{\text{temp}}(\mathbf{\Theta}) \tag{8}$$

where

$$E_{\text{lim}}(\mathbf{\Theta}) = \sum_{\theta_i \in \mathbf{\Theta}} \begin{cases} 0 & , \text{if } \theta_i^l \leq \theta_i \leq \theta_i^u \\ (\theta_i - \theta_i^l)^2 & , \text{if } \theta_i < \theta_i^l \\ (\theta_i^u - \theta_i)^2 & , \text{if } \theta_i > \theta_i^u \end{cases} \tag{9}$$

is a soft prior to enforce biomechanical pose plausibility, with $\mathbf{\Theta}^l, \mathbf{\Theta}^u$ being the lower and upper joint angle limits, respectively, and

$$E_{\text{temp}}(\mathbf{\Theta}) = ||\nabla\mathbf{\Theta} - \nabla\mathbf{\Theta}^{(t-1)}||_2^2 \tag{10}$$

enforces constant velocity to prevent dramatic pose changes. We empirically chose weights for the regularizers as: $w_l = 0.03$ and $w_t = 10^{-3}$. We optimize our objective using 20 iterations of conditioned gradient descent.

6. Results and Evaluation

We conducted several experiments to evaluate our method and different components of it. To facilitate evaluation, we captured a new benchmark dataset *EgoDexter* consisting of 3190 frames of natural hand interactions with objects in real cluttered scenes, moving egocentric viewpoints, complex hand-object interactions, and natural lighting. Of these, we manually annotated 1485 frames using an annotation tool to mark 2D and 3D fingertip positions, a common approach used in free hand tracking [1, 28]. In total we gathered 4 sequences (`Rotunda`, `Desk`, `Kitchen`, `Fruits`) featuring 4 different users (2 female), skin color variation, background variation, different objects, and camera motion. Note that the objects in *EgoDexter* are distinct from the objects in the *SynthHands* training data to show the ability of our approach to generalize. In addition, to enable evaluation of the different components of our method, we also held out a test set consisting of 5120 fully annotated frames from the *SynthHands* dataset.

Component Evaluation: We first analyze the performance of *HALNet* and *JORNet* on the synthetic test set. The main

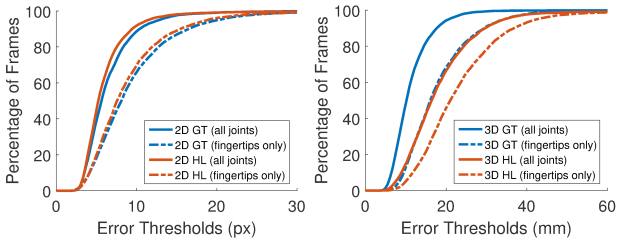


Figure 6: Comparison of 2D (left) and 3D (right) error of the joint position estimates of *JORNet*. *JORNet* was initialized with either the ground truth (GT, blue) or with the proposed hand localization step (HL, orange). We observe that HL initialization does not substantially reduce the performance of *JORNet*. As expected, fingertips-only errors (dashed lines) are higher than the errors for all joints.

goal of *HALNet* is to accurately localize the 2D position of the root (which either lies on the hand or on an occluder in front) accurately. We thus use 2D Euclidean pixel error between the ground truth root position and the predicted position as the evaluation metric. On average, *HALNet* produces an error of **2.2 px** with a standard deviation of 1.5 px on the test set. This low average error ensures that we always obtain reliable crops for *JORNet*.

To evaluate *JORNet*, we use the 3D Euclidean distance between ground truth joint positions (of all hand joints) and the predicted position as the error metric. For comparison, we also report the errors for only the 3D fingertip positions which are a stricter measure of performance. Since the output of *JORNet* is dependent on the crop estimated in the hand localization step, we evaluate two conditions: (1) using ground truth crops, (2) using crops from the hand localization step. This helps evaluate how hand localization affects the final joint positions. Figure 6 shows the percentage of the test set that produces a certain 2D or 3D error for all joints and fingertips only. For 3D error, we see that using ground truth (GT) crops is better than using the crops from the hand localization (HL). The difference is not substantial which shows that the hand localization step does not lead to catastrophic failures of *JORNet*. For 2D error, however, we observe that *JORNet* initialized with HL results in marginally better accuracy. We hypothesize that this is because *JORNet* is trained on noisy root positions (Section 4.4) while the ground truth lacks any such noise.

CNN Structure Evaluation: We now show that, on our real annotated benchmark *EgoDexter*, our approach that uses two subsequently applied CNNs is better than a single CNN to directly regress joint positions in cluttered scenes. We trained a CNN with the same architecture as *JORNet* but with the task of directly regressing 3D joint positions from full frame RGB-D images which often have large occlusions and scene clutter. In Figure 7, we show the 3D

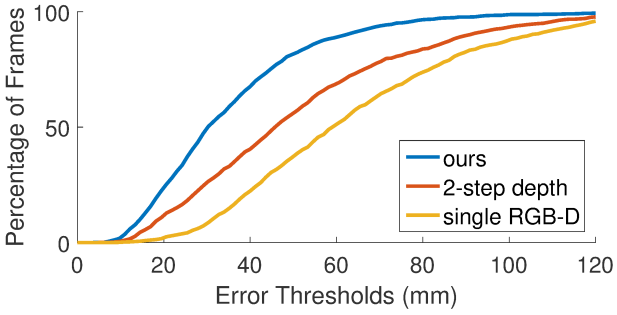


Figure 7: Comparison of our two-step RGB-D CNN architecture, the corresponding depth-only version and a single combined CNN which is trained to directly regress global 3D pose. Our proposed approach achieves the best performance on the real test sequences.

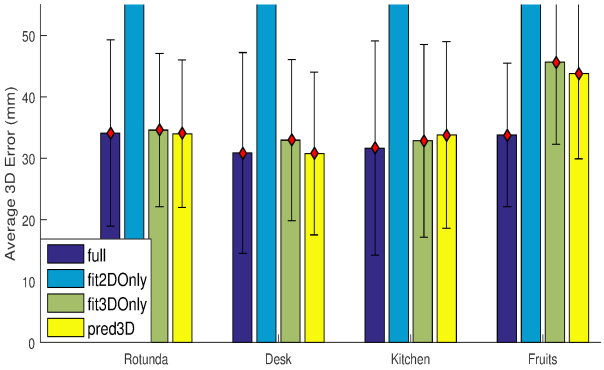


Figure 8: Ablative analysis of the proposed kinematic pose tracking on our real annotated dataset *EgoDexter* (average fingertip error). Using only the 2D fitting energy leads to catastrophic tracking failure on all sequences. The version restricted to the 3D fitting term achieves a similar error as the raw 3D predictions while it ensures biomechanical plausibility and temporal smoothness. Our full formulation that combines 2D as well as 3D terms yields the lowest error.

fingertip error plot for this CNN (single RGB-D) which is worse that our two-step approach. This shows that learning to directly regress 3D pose in cluttered scenes with occlusion is a harder task, which our approach simplifies by breaking it into two steps.

Input Data Evaluation: We next show, on our *EgoDexter* dataset, that using both RGB and depth input (RGB-D) is superior to using only depth, even when using both our CNNs. Figure 7 compares the 3D fingertip error of a variant of our two-step approach trained with only depth data. We hypothesize that additional color cues help our approach perform significantly better.

Gain of Kinematic Model: Figure 8 shows an ablative analysis of our energy terms as well as the effect of kinematic pose tracking on the final pose estimate. Because we

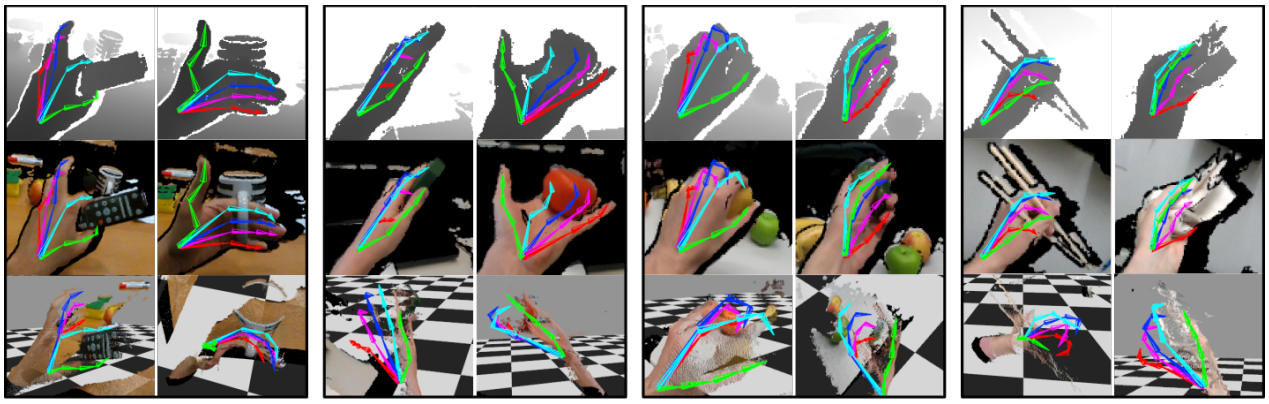


Figure 9: Qualitative results on our real annotated test sequences from the *EgoDexter* benchmark dataset. The results overlayed on the input images and the corresponding 3D view from a virtual viewpoint (bottom row) show that our approach is able to handle complex object interactions, strong self-occlusions and a variety of users and backgrounds.

enforce joint angle limits, temporal smoothness, and consistent bone lengths, our combined approach produces the lowest average error of **32.6 mm**.

We were unable to quantitatively evaluate on the only other existing *egocentric* hand dataset [20] due to a different sensor unsupported by our approach. To aid qualitative comparison, we include similar reenacted scenes, background clutter, and hand motion in the supplemental document and video.

Qualitative Results: Figure 9 shows qualitative results from our approach which works well for challenging real world scenes with clutter, hand-object interactions, and different hand shapes. We also show that a commercial solution (LeapMotion Orion) does not work well under severe occlusions caused by objects, see Figure 10 right. We refer to the supplemental document for results on how existing third person methods fail on *EgoDexter* and how our approach in fact generalizes to third person views.

Runtime Performance: Our entire method runs in real-time on an Intel Xeon E5-2637 CPU (3.5 GHz) with an Nvidia Titan X (Pascal). Hand localization takes 11 ms, 3D joint regression takes 6 ms, and kinematic pose tracking takes 1 ms.

Limitations: Our method works well even in challenging egocentric viewpoints and notable occlusions. However, there are some failure cases which are shown in Figure 10. Please see the supplemental document for a more detailed discussion of failure cases. We used large amounts of synthetic data for training our CNNs and simulated sensor noise for a specific camera preventing generalization. In the future, we would like to explore the application of deep domain adaptation [3] which offers a way to jointly make use of labeled *synthetic* data together with unlabeled or partially labeled *real* data.

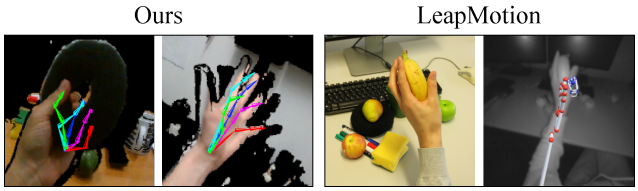


Figure 10: Fast motion that leads to misalignment in the colored depth image or failures in the hand localization step can lead to incorrect predictions (left two columns). LeapMotion Orion fails under large occlusions (right).

7. Conclusion

We have presented a method for hand pose estimation in challenging first-person viewpoints with large occlusions and scene clutter. Our method uses two CNNs to localize and estimate, in real time, the 3D joint locations of the hand. A pose tracking energy further refines the pose by estimating the joint angles of a kinematic skeleton for temporal smoothness. To train the CNNs, we presented *SynthHands*, a new photorealistic dataset that uses a merged reality approach to capture natural hand interactions, hand shape, size and color variations, object occlusions, and background variations from egocentric viewpoints. We also introduce a new benchmark dataset *EgoDexter* that contains annotated sequences of challenging cluttered scenes as seen from egocentric viewpoints. Quantitative and qualitative evaluation shows that our approach is capable of achieving low errors and consistent performance even under difficult occlusions, scene clutter, and background changes.

Acknowledgements: This work was supported by the ERC Starting Grant CapReal (335545). Dan Casas was supported by a Marie Curie Individual Fellow, grant 707326.

References

[1] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys. Motion Capture of Hands in Action using Discriminative Salient Points. In *European Conference on Computer Vision (ECCV)*, 2012. 2, 6

[2] A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. 4

[3] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. 8

[4] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3D Hand Pose Estimation in Single Depth Images: from Single-View CNN to Multi-View CNNs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[5] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *Computer Vision, 2009 IEEE 12th International Conference On*, pages 1475–1482. IEEE, 2009. 2

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4

[7] IntelRealSenseSR300. https://click.intel.com/realsense.html, 2016. 3

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[9] C. Keskin, F. Kra¸, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1228–1234, 2011. 2

[10] N. Kyriazis and A. Argyros. Scalable 3d tracking of multiple interacting objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3430–3437, 2014. 2

[11] LeapMotion. https://developer.leapmotion.com/orion, 2016. 2, 6

[12] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit. Efficiently Creating 3D Training Data for Fine Hand Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4

[13] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3316–3324, 2015. 1, 2

[14] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011. 2

[15] 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 *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2088–2095. IEEE, 2011. 2

[16] P. Panteleris, N. Kyriazis, and A. A. Argyros. 3d tracking of human hands in interaction with unknown objects. In *BMVC*, pages 123–1, 2015. 2

[17] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and Robust Hand Tracking from Depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1113, 2014. 1, 2, 3

[18] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, 35(6):162, 2016. 2

[19] G. Rogez, M. Khademi, J. Supančič III, J. M. M. Montiel, and D. Ramanan. 3D hand pose detection in egocentric RGB-D images. In *Workshop at the European Conference on Computer Vision*, pages 356–371. Springer, 2014. 2, 4

[20] G. Rogez, J. S. Supancic, and D. Ramanan. First-person pose recognition using egocentric workspaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4325–4333, 2015. 2, 4, 5, 8

[21] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1194–1201, 2012. 1

[22] J. Romero, H. Kjellström, and D. Kragic. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 458–463, 2010. 2

[23] 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 (IJCV)*, 115(3):211–252, 2015. 6

[24] A. Sinha, C. Choi, and K. Ramani. Deephand: robust hand pose estimation by completing a matrix imputed with deep features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4150–4158, 2016. 1, 2

[25] S. Sridhar, A. M. Feit, C. Theobalt, and A. Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. In *ACM Conference on Human Factors in Computing Systems*, pages 3643–3652, 2015. 1

[26] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and Robust Hand Tracking Using Detection-Guided Optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 3, 5

[27] 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 *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[28] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2456–2463, 2013. 2, 6

[29] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015. 2

[30] A. Tagliasacchi, M. Schroeder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust Articulated-ICP for Real-

Time Hand Tracking. *Computer Graphics Forum (Symposium on Geometry Processing)*, 34(5), 2015. 1, 2

[31] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014. 2

[32] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proc. ICCV*, 2015. 2

[33] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 35(4):143, 2016. 1, 2, 3

[34] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33, August 2014. 1, 2

[35] 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 (IJCV)*, 2016. 2

[36] D. Tzionas and J. Gall. 3d object reconstruction from hand-object interactions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 729–737, 2015. 2

[37] C. Wan, T. Probst, L. Van Gool, and A. Yao. Crossing nets: Dual generative models with a shared latent space for hand pose estimation. *arXiv preprint arXiv:1702.03431*, 2017. 1

[38] C. Wan, A. Yao, and L. Van Gool. Hand pose estimation from local surface normals. In *European Conference on Computer Vision*, pages 554–569. Springer, 2016. 2

[39] R. Wang, S. Paris, and J. Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proc. of UIST*, pages 549–558. ACM, 2011. 2

[40] C. Xu and L. Cheng. Efficient hand pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3462, 2013. 2

[41] Q. Ye, S. Yuan, and T.-K. Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 346–361. Springer, 2016. 2

[42] W. Zhao, J. Zhang, J. Min, and J. Chai. Robust realtime physics-based motion control for human grasping. *ACM Transactions on Graphics (TOG)*, 32(6):207, 2013. 1

[43] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. In *IJCAI*, 2016. 2