Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data

Yuxiao Zhou[1] Marc Habermann[2,3] Weipeng Xu[2,3] Ikhsanul Habibie[2,3] Christian Theobalt[2,3] Feng Xu*[1]

[1]BNRist and School of Software, Tsinghua University, [2]Max Planck Institute for Informatics, [3]Saarland Informatics Campus

Abstract

We present a novel method for monocular hand shape and pose estimation at unprecedented runtime performance of 100fps and at state-of-the-art accuracy. This is enabled by a new learning based architecture designed such that it can make use of all the sources of available hand training data: image data with either 2D or 3D annotations, as well as stand-alone 3D animations without corresponding image data. It features a 3D hand joint detection module and an inverse kinematics module which regresses not only 3D joint positions but also maps them to joint rotations in a single feed-forward pass. This output makes the method more directly usable for applications in computer vision and graphics compared to only regressing 3D joint positions. We demonstrate that our architectural design leads to a significant quantitative and qualitative improvement over the state of the art on several challenging benchmarks. Our model is publicly available for future research.[1]

1. Introduction

Hands are the most relevant tools for humans to interact with the real world. Therefore, capturing hand motion is of outstanding importance for a variety of applications in AR/VR, human computer interaction, and many more. Ideally, such a capture system should run at real time to provide direct feedback to the user, it should only leverage a single RGB camera to reduce cost and power consumption, and it should predict joint angles as they are more directly usable for most common applications in computer graphics, AR, and VR. 3D hand motion capture is very challenging, especially from a single RGB image, due to the inherent depth ambiguity of the monocular setting, self occlusions, complex and fast movements of the hand, and uniform skin appearance. The existing state-of-the-art methods resort to deep learning and have achieved significant improvement in recent years [2, 10, 51, 17, 3]. However, we observe two main problems with those methods.

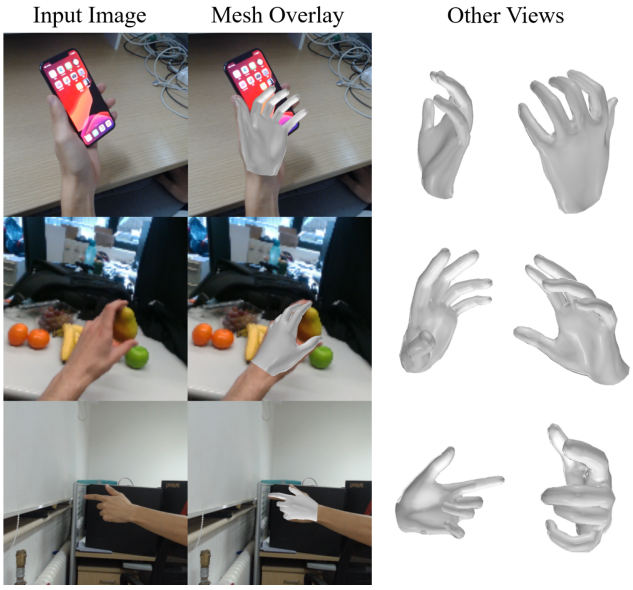

Figure 1. We present a novel hand motion capture approach that estimates 3D hand joint locations and rotations at real time from a single RGB image. Hand mesh models can then be animated with the predicted joint rotations. Our system is robust to challenging scenarios such as object occlusions, self occlusions, and unconstrained scale.

First, none of the existing methods makes use of all publicly available training data modalities, even though the annotated hand data is severely limited due to the difficulty of collecting real human hand images with 3D annotations. Specifically, to obtain 3D annotations, a particular capture setup is required, e.g., leveraging stereo cameras [50] or a depth camera [36, 42, 38, 49], which prevents collecting diverse data at large scale. An alternative is synthetic datasets [23, 54]. However, models trained on synthetic images do not generalize well to real images due to the domain gap [23]. In contrast, 2D annotated internet images with larger variation [33] are easier to obtain. However, it is nearly im-

*This work was supported by the National Key R&D Program of China 2018YFA0704000, the NSFC (No.61822111, 61727808, 61671268), the Beijing Natural Science Foundation (JQ19015, L182052), and the ERC Consolidator Grant 4DRepLy (770784). Feng Xu is the corresponding author.

[1]https://github.com/CalciferZh/minimal-hand

possible to annotate them with the 3D ground truth. We notice that, there is another valuable data modality neglected by all previous works - hand motion capture (MoCap) data. These datasets usually have a large variation in hand poses, but lack the paired images, since they are typically collected using data gloves [12] or 3D scanners [30]. Therefore, the previous methods cannot use them to learn the mapping from images to hand poses.

Second, most previous methods focus on predicting 3D joint positions [48, 34, 17, 3, 54]. Although useful for some applications, this positional representation is not sufficient to animate hand mesh models in computer graphics, where joint rotations are typically required. Some works [23, 29, 42] overcome this issue by fitting a kinematic hand model to the sparse predictions as a separate step. This not only requires hand-crafted energy functions, but expensive iterative optimization also suffers from erroneous local convergence. Other works [51, 1, 2] directly regress joint angles from the RGB image. All of them are trained in a weakly-supervised manner (using differentiable kinematics functions and the 3D/2D positional loss) due to the lack of training images paired with joint rotation annotations. Therefore, the anatomic correctness of the poses cannot be guaranteed.

To this end, we propose a novel real-time monocular hand motion capture approach that not only estimates 2D and 3D joint locations, but also maps them directly to joint rotations. Our method is rigorously designed for the utilization of all aforementioned data modalities, including synthetic and real image datasets with either 2D and/or 3D annotations as well as non-image MoCap data, to maximize accuracy and stability. Specifically, our architecture comprises two modules, *DetNet* and the *IKNet*, which predict 2D/3D joint locations and joint rotations, respectively. The proposed DetNet is a multi-task neural network for 3D hand joint detection that can inherently leverage fully and weakly annotated images at the same time by explicitly formulating 2D joint detection as an auxiliary task. In this multi-task training, the model learns how to extract important features from real images leveraging 2D supervision, while predicting 3D joint locations can be purely learned from synthetic data. The 3D shape of the hand can then be estimated by fitting a parametric hand model [30] to the predicted joint locations. To obtain the joint rotation predictions, we present the novel data-driven end-to-end IKNet that tackles the inverse kinematics (IK) problem by taking the 3D joint predictions of DetNet as input and regressing the joint rotations. Our IKNet predicts the kinematic parameters in a single feed-forward pass at high speed and it avoids complicated and expensive model fitting. During training, we can incorporate MoCap data that provides direct rotational supervision, as well as 3D joint position data that provides weak positional supervision, to learn the pose priors and correct the errors in 3D joint predictions. In summary, our contributions are:

- A new learning based approach for monocular hand shape and motion capture, which enables the joint usage of 2D and 3D annotated image data as well as stand-alone motion capture data.
- An inverse kinematics network that maps 3D joint predictions to the more fundamental representation of joint angles in a single feed-forward pass and that allows joint training with both positional and rotational supervision.

Our method outperforms state-of-the-art methods, both quantitatively and qualitatively on challenging benchmarks, and shows unseen runtime performance.

2. Related Work

In the following, we discuss the methods that use a single camera to estimate 3D hand pose, which are closely related to our work.

Depth based Methods. Many works proposed to estimate hand pose from depth images due to the wide spread of commodity depth cameras. Early depth based works [28, 21, 31, 7, 37, 41] estimate hand pose by fitting a generative model onto a depth image. Some works [35, 32, 40, 36, 43] additionally leveraged discriminative predictions for initialization and regularization. Recently, deep learning methods have been applied to this area. As a pioneer work, Tompson et al. [42] proposed to used CNN in combination with randomized decision forests and inverse kinematics to estimate hand pose from a single depth image at real time. Follow-up works achieved better performance by utilizing priors and context [25], high-level knowledge [39], a feedback loop [26, 27], or intermediate dense guidance map supervision [46]. [52, 6] proposed to use several branches to predict the pose of each part, e.g. palm and fingers, and exploit cross-branch information. Joint estimation of hand shape and pose was also proposed [19]. Wan et al. [44] exploited unlabeled depth maps for self-supervised finetuning, while Mueller et al. [24] constructed a photorealistic dataset for better robustness. Some works leveraged other representations, such as point clouds [8, 11, 4, 18] and 3D voxels [16, 22, 9], which can be retrieved from depth images. Although these works achieve appealing results, they suffer from the inherent drawbacks of depth sensors, which do not work under bright sunlight, have a high power consumption and people have to be close to the sensor.

Monocular RGB Methods. To this end, people recently started to research 3D hand pose estimation from monocular RGB images, which is even more challenging than the depth based setting due to the depth ambiguity. Zimmermann and Brox [54] trained a CNN based model that estimates 3D joint coordinates directly from an RGB image.

Iqbal et al. [17] used a 2.5D heat map formulation, that encodes 2D joint locations together with depth information, leading to a large boost in accuracy. For better generalization, many works [3, 34, 48] utilized depth image datasets to enlarge the diversity seen during training. Mueller et al. [23] proposed a large scale rendered dataset post-processed by a CycleGAN[53] to bridge the domain gap. However, they only focused on joint position estimation but refrained from joint rotation recovery, which is much better for hand mesh animation. To estimate joint rotations, [47, 29] fitted a generic hand model to the predictions via an iterative optimization based approach, which is not time-efficient and requires hand-crafted energy functionals. [51, 1] proposed to regress the parameters of a deformable hand mesh model from the input image in an end-to-end manner. Nonetheless, the estimated rotations can only be weakly supervised, resulting in inferior accuracy. Ge et al. [10] directly regressed a hand mesh using a GraphCNN [5], but a special dataset with ground truth hand meshes is required, which is hard to construct. Their model-free method is also less robust to challenging scenes. In contrast, by fully exploiting existing datasets from different modalities, including image data and non-image MoCap data, our approach obtains favorable accuracy and robustness.

3. Method

As shown in Fig. 2, our method includes two main modules. First, the joint detection network, DetNet (Sec. 3.1), predicts 2D and 3D hand joint positions from a single RGB image under a multi-task scheme. Then, we can retrieve the shape of the hand by fitting a hand model to the 3D joint predictions (Sec. 3.2). Second, the inverse kinematics network, IKNet (Sec. 3.3), takes the 3D joint predictions and converts them into a joint rotation representation in an end-to-end manner.

3.1. Hand Joint Detection Network DetNet

The DetNet takes the single RGB image and outputs root-relative and scale-normalized 3D hand joint predictions as well as 2D joint predictions in image space. The architecture of DetNet comprises 3 components: a feature extractor, a 2D detector, and a 3D detector.

Feature Extractor. We use the backbone of the ResNet50 architecture [14] as our feature extractor where the weights are initialized with the Xavier initialization [13]. It takes images at a resolution of 128×128 and outputs a feature volume F of size $32 \times 32 \times 256$.

2D Detector. The 2D detector is a compact 2-layer CNN that takes the feature volume F and outputs heat maps H_j corresponding to the $J = 21$ joints. As in [45], a pixel in H_j encodes the confidence of that pixel being covered by joint j. The heat maps are used for 2D pose estimation, which is a subtask, supervised by ground truth 2D annotations. Thus, the feature extractor and the 2D detector can be trained with 2D labeled real image data from the internet. This drastically improves generalization ability since during training both feature extractor and 2D detector see in-the-wild images that contain more variations than images from 3D annotated datasets.

3D Detector. Now, the 3D detector takes the feature maps F and the heat maps H, and estimates 3D hand joint positions in the form of *location maps* L, similar to [20]. For each joint j, L_j has the same 2D resolution as H_j, and each pixel in L_j encodes joint j's 3D coordinates. This redundancy helps to the robustness. Similar to L, we also estimate delta maps D where each pixel in D_b encodes the orientation of bone b, represented by a 3D vector from the parent joint to the child joint. This intermediate representation is needed to explicitly inform the network about the relation of neighboring joints in the kinematic chain. In the 3D detector, we first use a 2-layer CNN to estimate the delta maps D from the heat maps H and feature maps F. Next, heat maps H, feature maps F, and delta maps D are concatenated and fed into another 2-layer CNN to obtain the final location maps L. The location maps L and the delta maps D are supervised by 3D annotations. During inference, the 3D position of joint j can be retrieved by a simple look-up in the location map L_j at the uv-coordinate corresponding to the maxima of the heat map H_j. To alleviate the fundamental depth-scale ambiguity in the monocular setting, the predicted coordinates are relative to a root joint and normalized by the length of a reference bone. We select the middle metacarpophalangeal to be the root joint, and the bone from this joint to the wrist is defined as the reference bone.

Loss Terms. Our loss function

$$\mathcal{L}_{\text{heat}} + \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{delta}} + \mathcal{L}_{\text{reg}} \tag{1}$$

comprises four terms to account for the multi-task learning scheme. First, $\mathcal{L}_{\text{heat}}$ is defined as

$$\mathcal{L}_{\text{heat}} = ||H^{\text{GT}} - H||_F^2 \tag{2}$$

which ensures that the regressed heatmaps H are close to the ground truth heatmaps H^{GT}. $||\cdot||_F$ denotes the Frobenius norm. To generate the ground truth heat maps H_j^{GT} for joint j, we smooth H_j^{GT} with a Gaussian filter centered at the 2D annotation using a standard deviation of $\sigma = 1$. Again note that $\mathcal{L}_{\text{heat}}$ only requires 2D annotated image datasets. We particularly stress the importance of such images, as they contain much more variation than those with 3D annotations. Thus, this loss supervises our feature extractor and our 2D detector to learn the important features for hand joint detection on in-the-wild images. To supervise the 3D detector, we propose two additional loss terms

$$\mathcal{L}_{\text{loc}} = ||H^{\text{GT}} \odot (L^{\text{GT}} - L)||_F^2 \tag{3}$$

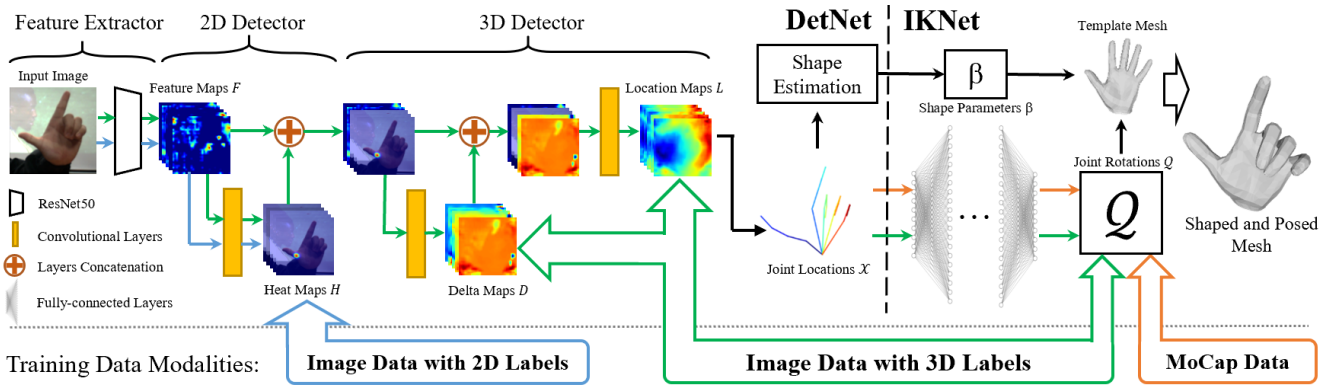

Figure 2. Overview of our architecture. It comprises two modules: first, our DetNet predicts the 2D and 3D joint positions from a single RGB image. Second, our IKNet takes the 3D joint predictions of DetNet and maps them to joint angles.

$$\mathcal{L}_{\text{delta}} = ||H^{\text{GT}} \odot (D^{\text{GT}} - D)||_F^2 \quad (4)$$

which measure the difference between ground truth and predicted location maps L and delta maps D, respectively. Ground truth location maps L^{GT} and delta maps D^{GT} are constructed by tiling the coordinates of the ground truth joint position and bone direction to the size of the heat maps. Since we are mainly interested in the 3D predictions at the maxima of the heat maps, the difference is weighted with H^{GT}, where $\odot$ is the element-wise matrix product. $\mathcal{L}_{\text{reg}}$ is a $L2$ regularizer for the weights of the network to prevent overfitting. During training, data with 2D and 3D annotations are mixed in the same batch, and all the components are trained jointly. Under this multi-task scheme, the network learns to predict 2D poses under diverse real world appearance from 2D labeled images, as well as 3D spatial information from 3D labeled data.

Global Translation. If the camera intrinsics matrix K and the reference bone length l_{ref} are provided, the absolute depth z_{r} of the root joint can be computed by solving

$$l_{\text{ref}} = ||K^{-1} z_{\text{r}} \begin{bmatrix} u_{\text{r}} \\ v_{\text{r}} \\ 1 \end{bmatrix} - K^{-1}(z_{\text{r}} + l_{\text{ref}} * d_{\text{w}}) \begin{bmatrix} u_{\text{w}} \\ v_{\text{w}} \\ 1 \end{bmatrix} ||_2 \quad (5)$$

Here, subscripts $\cdot_{\text{r}}$ and $\cdot_{\text{w}}$ denote the root and wrist joint, respectively. u and v are the 2D joint predictions in the image plane and d_{w} is the normalized and root-relative depth of the wrist regressed by DetNet. As z_{r} is the only unknown variable, one can solve for it in closed form. After computing z_{r}, the global translation in x and y dimension can be computed via the camera projection formula.

3.2. Hand Model and Shape Estimation

Hand Model. We choose MANO [30] as the hand model to be driven by the output of our IKNet. The surface mesh of MANO can be fully deformed and posed by the shape parameters $\beta \in \mathbb{R}^{10}$ and pose parameters $\theta \in \mathbb{R}^{21\times 3}$. More specifically, β represents the coefficients of a shape PCA bases which is learned from hand scans, while θ represents joint rotations in axis-angle representation. They allow to deform the mean template $\bar{T} \in \mathbb{R}^{V\times 3}$ to match the shape of different identities as well as to account for pose-dependent deformations. Here, V denotes the number of vertices. Before posing, the mean template $\bar{T}$ is deformed as

$$\mathcal{T}(\beta, \theta) = \bar{T} + \mathcal{B}_s(\beta) + \mathcal{B}_p(\theta) \quad (6)$$

where $\mathcal{B}_s(\beta)$ and $\mathcal{B}_p(\theta)$ are shape and pose blendshapes, respectively. Then the posed hand model $\mathcal{M}(\theta, \beta) \in \mathbb{R}^{V\times 3}$ is defined as

$$\mathcal{M}(\theta, \beta) = W(\mathcal{T}(\theta, \beta), \theta, \mathcal{W}, \mathcal{J}(\theta)) \quad (7)$$

where $W(\cdot)$ is a standard linear blend skinning function that takes the deformed template mesh $\mathcal{T}(\beta, \theta)$, pose parameters θ, skinning weights $\mathcal{W}$, and posed joint locations $\mathcal{J}(\theta)$.

Shape Estimation. Since we are not only interested in the pose of the hand but also its shape, we utilize the predicted joint positions to estimate shape parameters β of the MANO model. As the predictions are scale-normalized, the estimated shape can only represent relative hand shape, e.g., the ratio of fingers to palm. We compute the hand shape β by minimizing

$$E(\beta) = \sum_b || \frac{l_b(\beta)}{l_{\text{ref}}(\beta)} - l_b^{\text{pred}} ||_2^2 + \lambda_\beta ||\beta||_2^2. \quad (8)$$

Here, the first term ensures that for every bone b the bone length of the deformed hand model $l_b(\beta)$ matches the length of the predicted 3D bone length l_b^{pred}, that can be derived from the 3D predictions of DetNet. Label $\cdot_{\text{ref}}$ refers to the reference bone of the deformed MANO model. The second term acts as a $L2$ regularizer on the shape parameters and is weighted by λ_β.

3.3. Inverse Kinematics Network IKNet

Although 3D joint locations can explain the hand pose, such a representation is not sufficient to animate hand mesh

models, which is important for example in computer graphics (CG) applications. In contrast, a widely-used representation to drive CG characters are joint rotations. We therefore infer in the network joint rotations from joint locations, also known as the inverse kinematics (IK) problem. To this end, we propose a novel end-to-end neural network, IKNet, to solve the inverse kinematics problem. The main benefits of our *learning based* IKNet are: First, our design allows us to incorporate MoCap data as an additional data modality to provide full supervision during training. This is in stark contrast to methods that directly regress rotations from the image [51, 1, 2] which only allow weakly supervised training. Second, we can solve the IK problem at much higher speed since we only require a single feed-forward pass compared to iterative model fitting methods [23, 29]. Third, hand pose priors can be directly learned from the data in contrast to hand-crafted priors in optimization based IK [23, 29]. Finally, we also show that our IKNet can correct noisy 3D predictions of DetNet and the joint rotation representation is by nature bone-scale preserving. The similar idea of an IK network was also proposed in [15], but was used for denoising marker-based MoCap data, while we perform hand pose estimation.

MoCap Data. When it comes to training the IKNet, one ideally wants to have paired samples of 3D hand joint positions and the corresponding joint rotation angles. The MANO model comes with a dataset that contains 1554 poses of real human hands from 31 subjects. Originally, the rotations are in the axis-angle representation and we convert them to the quaternion representation, which makes interpolation between two poses easier. However, this dataset alone would still not contain enough pose variations. Therefore, we augment the dataset based on two assumptions: *1)* we assume the pose of each finger is independent of other fingers; *2)* any interpolation in quaternion space from the rest pose to a pose from the extended dataset, that is based on 1), is valid. Based on *1)*, we choose independent poses for each finger from the original dataset and combine them to form unseen hand poses. Based on *2)*, we can now interpolate between the rest pose and the new hand poses. To account for different hand shapes, we also enrich the dataset by sampling β with the normal distribution $\mathcal{N}(0, 3)$. Following the above augmentation technique, we produce paired joint location and rotation samples on-the-fly during training.

3DPosData. However, if we train IKNet purely on this data, it is not robust with respect to the noise and errors that are contained in the 3D predictions of DetNet. This is caused by the fact that the paired MoCap data is basically noiseless. Therefore, we also leverage the 3D annotated image data. In particular, we let the pre-trained DetNet produce the 3D joint predictions for all the training examples with 3D annotations and use those joint predictions as the input to the IKNet. The estimated joint rotations of the IKNet are then passed through a forward kinematic layer to reconstruct the joint positions, which are then supervised by the corresponding ground truth 3D joint annotations. In other words, we additionally construct a dataset with paired 3D DetNet predictions and ground truth 3D joint positions, which is used as a weak supervision to train the IKNet. We refer to this dataset as 3DPosData in the following. In this way, the IKNet learns to handle the 3D predictions of DetNet and is robust to noisy input.

Network Design. We design the IKNet as a 7-layer fully-connected neural network with batch normalization, and use sigmoid as the activation function except for the last layer that uses a linear activation. We encode the input 3D joint positions as $\mathcal{I} = [\mathcal{X}, \mathcal{D}, \mathcal{X}_{\text{ref}}, \mathcal{D}_{\text{ref}}] \in \mathbb{R}^{4\times J\times 3}$, where $\mathcal{X}$ are the root-relative scale-normalized 3D joint positions as in Sec. 3.1; $\mathcal{D}$ is the orientation of each bone, which we additionally provide as input to explicitly encode information of neighboring joints. $\mathcal{X}_{\text{ref}}$, $\mathcal{D}_{\text{ref}}$ encode information about the shape identity and are defined as the 3D joint positions and bone orientations in the rest pose, respectively. They can be measured in advance for better accuracy, or inferred from the predictions of the DetNet, as described in Sec.3.2. The output of the IKNet is the global rotation of each joint represented as a quaternion $\hat{\mathcal{Q}} \in \mathbb{R}^{J\times 4}$, which is then normalized to be a unit quaternion $\mathcal{Q}$. We prefer the quaternion representation over an axis angle one due to the better interpolation properties that are required in our data augmentation step. Additionally, quaternions can be converted to rotation matrices, as later used in our losses, without using trigonometric functions which are more difficult to train since they are non-injective. To apply the final pose to the MANO model, we convert the quaternions $\mathcal{Q}$ back to the axis-angle representation, and then deform the model according to Eq. 7.

Loss Terms. Our loss function comprises four terms

$$\mathcal{L}_{\cos} + \mathcal{L}_{\text{l2}} + \mathcal{L}_{\text{xyz}} + \mathcal{L}_{\text{norm}}. \tag{9}$$

First, $\mathcal{L}_{\cos}$ measures the distance between the cosine value of the difference angle, which is spanned by the ground truth quaternion $\mathcal{Q}^{\text{GT}}$ and our prediction $\mathcal{Q}$, as

$$\mathcal{L}_{\cos} = 1 - real(\mathcal{Q}^{\text{GT}} * \mathcal{Q}^{-1}). \tag{10}$$

$real(\cdot)$ takes the real part of the quaternion, $*$ is the quaternion product, and $\mathcal{Q}^{-1}$ is the inverse of quaternion $\mathcal{Q}$. Further, $\mathcal{L}_{\text{l2}}$ directly supervises the predicted quaternion $\mathcal{Q}$:

$$\mathcal{L}_{\text{l2}} = ||\mathcal{Q}^{\text{GT}} - \mathcal{Q}||_2^2. \tag{11}$$

The proposed two losses can only be applied on the MoCap data. To also use 3DPosData, we propose a third loss, $\mathcal{L}_{\text{xyz}}$, to measure the error in terms of 3D coordinates after posing

$$\mathcal{L}_{\text{xyz}} = ||\mathcal{X}^{\text{GT}} - FK(\mathcal{Q})||_2^2 \tag{12}$$

where $FK(\cdot)$ refers to the forward kinematics function and $\mathcal{X}^{\mathrm{GT}}$ is the ground truth 3D joint annotation. Finally, to softly constrain the un-normalized output $\hat{\mathcal{Q}}$ to be unit quaternions, we apply $\mathcal{L}_{\mathrm{norm}}$ as

$$\mathcal{L}_{\mathrm{norm}} = |1 - ||\hat{\mathcal{Q}}||_2^2|. \tag{13}$$

4. Results

In this section, we first provide implementation details (Sec. 4.1). Then, we show qualitative results on challenging examples (Sec. 4.2). Finally, we compare our method to previous work (Sec. 4.3) and perform an ablation study to evaluate the importance of all our design choices (Sec. 4.4).

4.1. Implementation Details

All our experiments are performed on a machine with NVIDIA GTX1080Ti graphics card, where DetNet takes $8.9ms$ and IKNet takes $0.9ms$ for a single feed-forward pass. Thus, we achieve a state-of-the-art runtime performance of over $100fps$.

Training Data. Our DetNet is trained on 3 datasets: the CMU Panoptic Dataset (CMU) [33], the Rendered Handpose Dataset (RHD) [54] and the GANerated Hands Dataset (GAN) [23]. The CMU dataset contains 16720 image samples with 2D annotations gathered from real world. RHD and GAN are both synthetic datasets that contain 41258 and 330000 images with 3D annotations, respectively. Note that DetNet is trained without any real images with 3D annotations. We found that the real image 3D datasets do not contain enough variations and let our network overfit resulting in poor generalization across different datasets. To train the IKNet, we leverage the MoCap data from the MANO model and the 3DPosData, as discussed before.

4.2. Qualitative Results

In Fig. 3, we show results of our novel method on several challenging in-the-wild images demonstrating that it generalizes well to unseen data. Most importantly, we not only predict 3D joint positions but also joint angles, allowing us to animate a hand surface model directly. Such an output representation is much more useful in many applications in graphics and vision. Further Fig. 3 demonstrates that our method works well for very fast motions and blurred images (top left), as well as complex poses such as grasping (bottom left). Occlusions by objects (top right), self occlusions and challenging view points (bottom right) can also be handled. More results are shown in our supplemental material. In Fig. 4, we demonstrate that our approach can capture different hand shapes just from a single image. Note that finger and palm shape are correctly adjusted and they look plausible. In Fig 5, we qualitatively compare our results to Zimmermann and Brox [54] and Ge et al. [10] on challenging images. While [54] only recovers 3D joint positions, our method can animate a full 3D hand mesh model due to the joint rotation representation. We also demonstrate superior robustness compared to [10], which we attribute to the combined training on 2D labeled in-the-wild images and the MoCap data.

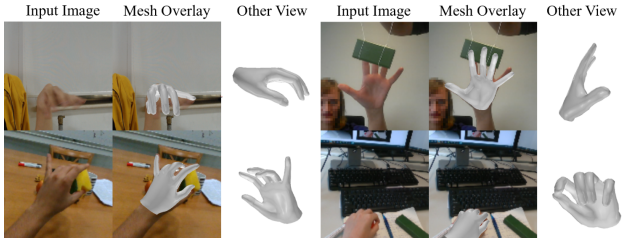

Figure 3. We demonstrate our results under several challenging scenarios: motion blur, object occlusion, complex pose, and unconstrained viewpoint. We show our results overlayed onto the input image and from a different virtual camera view.

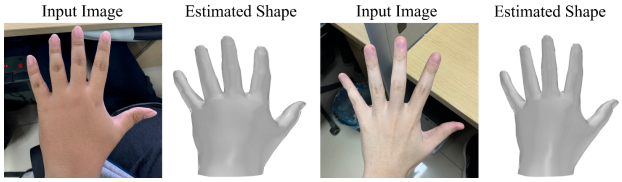

Figure 4. Illustration of our shape results. Note that our recovered shapes look visually plausible and reflect the overall shape of the subject's hand in the input image.

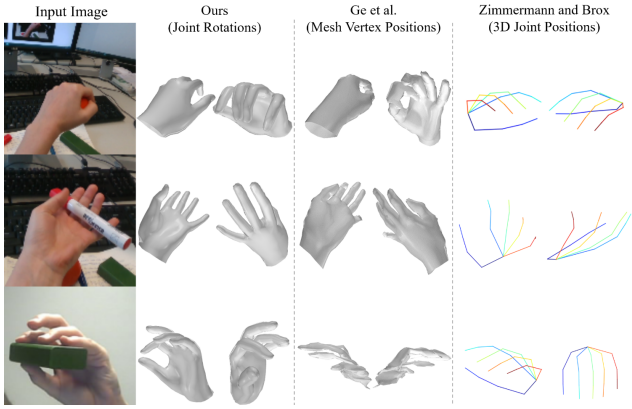

Figure 5. Comparison with [54] and Ge et al. [10]. Our approach cannot only output a fully deformed and posed dense 3D hand model, but also shows better robustness under occlusions compared to previous work. We show the same pose rendered from original and different camera view.

4.3. Comparison to Related Work

Evaluation Datasets and Metrics. We evaluate our approach on four public datasets: the test sets of RHD [54] and Stereo Hand Pose Tracking Benchmark (STB) [50], Dexter+Object (DO) [36] and EgoDexter (ED) [24]. Again, note that RHD is a synthetic dataset. The STB dataset contains 12 sequences of a unique subject with 18000 frames in total. Following [23], we evaluate our model on 2 se-

quences. The DO dataset comprises 6 sequences of 2 subjects interacting with objects from third view. The ED dataset is composed of 4 sequences of 2 subjects performing hand-object interactions in the presence of occlusions captured from an egocentric view. We use the following evaluation metrics: the percentage of correct 3D keypoints (PCK), and the area under the PCK curve (AUC) with thresholds ranging from $20mm$ to $50mm$. As previous work, we perform a global alignment to better measure the local hand pose. For ED and DO we aligned the centroid of the finger tip predictions to the GT one; for RHD and STB we aligned our root to the ground truth root location.

Quantitative Comparison. In Table. 1, we compare our approach to other state-of-the-art methods. Note that *not* all the methods were trained on the exact same data. Some methods use additional data, some of which not publicly available, for higher accuracy, including: synthetic images with ground truth hand mesh [10], depth images [48, 3, 34], real images with 2D annotations [17], and real images with 3D labels from a panoptic stereo [47]. We argue among all test datasets, the most fair comparison can be reported on the DO and ED dataset since no model used them for training. This further means that the evaluation on DO and ED gives a good estimate of how well models generalize. On DO and ED, our approach outperforms others by a large margin. This is due to our novel architecture that allows combining all available data modalities, including 2D and 3D annotated image datasets as well as MoCap data. We further stress the importance of the dataset combination used to train our model.

On STB, our accuracy is within the range of our results on DO and ED, further proving that our approach generalizes across datasets. While we achieve a worse accuracy on STB compared to others, note that our final model is *not* trained on STB in contrast to all other approaches. As many works have mentioned [47, 48, 51, 17], the STB dataset is easily saturated. Models tend to overfit to STB due to its large amount of frames and little variation. We argue that the utilization of STB for training would make the training data imbalanced and harm the generalization. This is evidenced by our additional experiment where we add STB to our training set and achieve an AUC of 0.991 on the test set of STB which is on par with previous work, but this model suffers from a huge performance drop on all other three benchmarks. Therefore, we did not use STB to train our final model.

For RHD, again our model achieves a consistent result as on other benchmarks. As a synthetic dataset, RHD has different appearance and pose distribution compared to real datasets. Previous work accounts for this by exclusively training or fine-tuning on RHD leading to superior results. Our final model avoids this since generalization to real images is harmed. To still proof that our architectural design is on par or better than state-of-the-art models, we made another evaluation where also exclusively train on RHD and achieve an AUC of 0.893 that is in the same ball park with others.

Method	AUC of PCK			
	DO	ED	STB	RHD
Ours	**.948**	**.811**	.898	.856*
Ge et al. [10]	-	-	**.998***	.920*
Zhang et al. [51]	.825	-	.995*	.901*
Yang et al. [48]	-	-	.996*	**.943***
Baek et al. [1]	.650	-	.995*	.926*
Xiang et al. [47]	.912	-	.994*	-
Boukhayma et al. [2]	.763	.674	.994*	-
Iqbal et al. [17]	.672	.543	.994*	-
Cai et al. [3]	-	-	.994*	.887*
Spurr et al. [34]	.511	-	.986*	.849*
Mueller et al. [23]	.482	-	.965*	-
Z&B [54]	.573	-	.948*	.670*

Table 1. Comparison with state-of-the-art methods on four public datasets. We use "*" to note that the model was trained on the dataset, and use "-" for those who did not report the results. Our system outperforms others by a large margin on the DO and ED dataset which we argue is the most fair comparison as none of the models are trained on these datasets. As [17] only reports results without alignment, we report the absolute values for this method.

	Variants of our Method	AUC of PCK		
		DO	ED	STB
1)	**Ours**	**.948**	.811	**.898**
2)	w/o IKNet	.923	.804	.891
3)	w/o L_{l2} and $L_{\cos}$	.933	.823	.869
4)	w/o 3DPosData	.926	.809	.873
5)	w/o L_{l2}	.943	**.812**	.890
5)	w/o $L_{\cos}$	.840	.782	.808

Table 2. Ablation study. We evaluate the influence of: *2)* IKNet *3)* Direct rotational supervision on joint rotations. *4)* Weak supervision on joint rotations. *5)* Loss terms on the quaternions.

4.4. Ablation Study

In Table. 2 and Fig. 6, we evaluate the key components of our approach: specifically, we evaluate *1)* our architectural design and the combination of training data compared to a baseline, *2)* the impact of the IKNet over a pure 3D joint position regression of DetNet, which we refer to as DetNet-only, *3)* the influence of direct rotational supervision on joint rotations enabled by the MoCap data, *4)* how our weak supervision, using the 3DPosData, helps the IKNet to adapt to noisy 3D joint predictions, and *5)* the influence of the two loss terms on the quaternions. *1)* As a baseline, we compare to Zhang et al. [51] as they report state-of-the-art results on DO without using any datasets that are not pub-

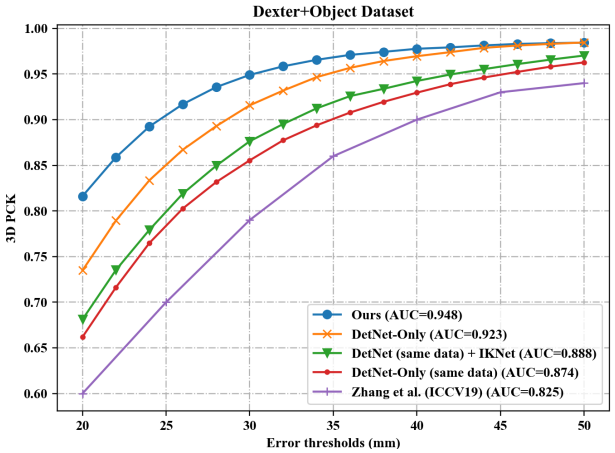

Figure 6. Ablation study for the training data on DO. We use "same data" to indicate our model trained with RHD and STB, which is the same as Zhang et al. [51]. We demonstrate that our architecture is superior to theirs by design. Integrating more data further boosts the results.

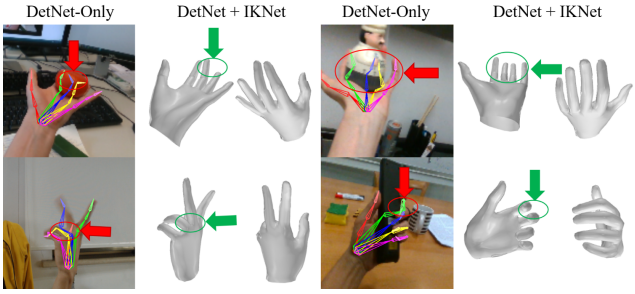

Figure 7. Our IKNet is able to compensate some errors from the DetNet based on the prior learned from the MoCap data.

licly available (in contrast to Xiang et al. [47] who leverage 3D annotations on CMU that are not released). To evaluate our model architecture, we trained DetNet on exactly the same data as [51] which brings an improvement of around 5% compared to [51]. This shows that our architecture itself helps to improve accuracy. Adding the IKNet, additionally trained on MoCap data, further improves the result. This further proves that disentangling the motion capture task into joint detection and rotation recovery makes the model easier to train, and also enables to leverage of MoCap data. Finally, the results are significantly improved with the proposed combination of training data, especially the in-the-wild 2D-labeled images. *2)* Across all datasets, IKNet improves the over DetNet-only. This can be explained as our IKNet acts like a pose prior, learned from MoCap data, and can therefore correct raw 3D joint predictions of DetNet. In Fig. 7, the DetNet itself cannot estimate the 3D joint positions correctly. Nevertheless, our learned hand pose prior, built-in the IKNet, can correct those wrong predictions. *3)* Here, we removed all rotational supervision terms and only use weak supervision. Despite the numerical results are on par with our final approach, the estimated rotations are

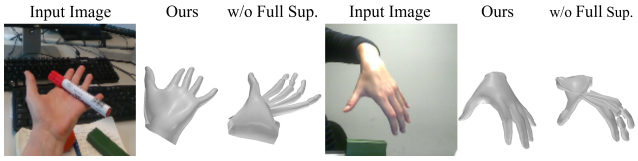

Figure 8. Comparison between IKNets with and without rotational supervision from MoCap data. Note that even though 3D joint positions match the ground truth, without this supervision unnatural poses are estimated.

anatomically wrong, as shown in Fig 8. This indicates that adding rotational supervision, retrieved from the MoCap data, makes training much easier and leads to anatomically more correct results. *4)* The difference between 1) and 4) in Table. 2 demonstrates that the 3DPosData is crucial to make the IKNet compatible to the DetNet. In order words, without this data the IKNet never sees the noisy 3D predictions of the DetNet but only the accurate 3D MoCap data. Thus, it even makes the results worse. Feeding the IKNet with the output of the pre-trained DetNet helps to deal with the noisy 3D predictions and achieves the best results. *5)* Finally in terms of network training, we found that $L_{\cos}$ is a better metric to measure the difference between two quaternions compared to the naive L_{12} and the combination of the two gives the highest accuracy on average.

5. Conclusion

We proposed the first learning based approach for monocular hand pose and shape estimation that utilizes data from two completely different modalities: image data and MoCap data. Our new neural network architecture features a trained inverse kinematics network that directly regresses joint rotations. These two aspects leads to a significant improvement over state of the art in terms of accuracy, robustness and runtime. In the future, we plan to extend our model to capture the hand texture by incooperating dense 3D scans. Another direction is the joint capturing of two interacting hands from a single RGB image which currently is only possible with depth sensors.

References

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Adnane Boukhayma, Rodrigo de Bem, and Philip H.S. Torr. 3d hand shape and pose from images in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *The European Conference on Computer Vision (ECCV)*, 2018.

[4] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[6] Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. Crossinfonet: Multi-task information sharing based hand pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Shachar Fleishman, Mark Kliger, Alon Lerner, and Gershom Kutliroff. Icpik: Inverse kinematics based articulated-icp. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2015.

[8] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Real-time 3d hand pose estimation with 3d convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):956–970, April 2019.

[10] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[12] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Trans. Graph.*, 38(4):41:1–41:15, July 2019.

[13] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] Daniel Holden. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[16] Fuyang Huang, Ailing Zeng, Minhao Liu, Jing Qin, and Qiang Xu. Structure-aware 3d hourglass network for hand pose estimation from single depth image. In *The British Machine Vision Conference (BMVC)*, 2018.

[17] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *The European Conference on Computer Vision (ECCV)*, 2018.

[18] Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Héloir, and Didier Stricker. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *The International Conference on 3D Vision (3DV)*, 2018.

[20] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics*, 36(4), July 2017.

[21] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, pages 63–70. Canadian Information Processing Society, 2013.

[22] Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

[25] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. In *Computer Vision Winter Workshop*, 2015.

[26] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[27] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Generalized feedback loop for joint hand-object pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[28] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *The British Machine Vision Conference (BMVC)*, 2011.

[29] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision*, 2018.

[30] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, Nov. 2017.

[31] Matthias Schröder, Jonathan Maycock, Helge Ritter, and Mario Botsch. Real-time hand tracking using synergistic inverse kinematics. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014.

[32] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3633–3642, New York, NY, USA, 2015. ACM.

[33] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[35] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[36] Srinath Sridhar, Franziska Mueller, Michael Zollhoefer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *The European Conference on Computer Vision (ECCV)*, 2016.

[37] Andrea Tagliasacchi, Matthias Schroeder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. *Computer Graphics Forum (Symposium on Geometry Processing)*, 34(5), 2015.

[38] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[39] D. Tang, Q. Ye, S. Yuan, J. Taylor, P. Kohli, C. Keskin, T. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for hand pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2161–2175, Sep. 2019.

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

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

[42] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169:1–169:10, Sept. 2014.

[43] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision (IJCV)*, 2016.

[44] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3d hand pose estimation through training by fitting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[45] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[46] Xiaokun Wu, Daniel Finnegan, Eamonn O'Neill, and Yong-Liang Yang. Handmap: Robust hand pose estimation via intermediate dense guidance map supervision. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[47] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[48] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[49] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[50] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. In *The IEEE International Conference on Image Processing (ICIP)*, 2017.

[51] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[52] Yidan Zhou, Jian Lu, Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. Hbe: Hand branch ensemble network for real-time 3d hand pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[53] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[54] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.