Monocular RGB Hand Pose Inference from Unsupervised Refinable Nets

Endri Dibra, Silvan Melchior, Ali Balkis, Thomas Wolf, Cengiz Öztireli, Markus Gross
Department of Computer Science, ETH Zürich
{{edibra,cengizo,grossm}@inf, {silvanm,abalkis,wolftho}@student}.ethz.ch

Abstract

3D hand pose inference from monocular RGB data is a challenging problem. CNN-based approaches have shown great promise in tackling this problem. However, such approaches are data-hungry, and obtaining real labeled training hand data is very hard. To overcome this, in this work, we propose a new, large, realistically rendered hand dataset and a neural network trained on it, with the ability to refine itself unsupervised on real unlabeled RGB images, given corresponding depth images. We benchmark and validate our method on existing and captured datasets, demonstrating that we strongly compare to or outperform state-of-the-art methods for various tasks ranging from 3D pose estimation to hand gesture recognition.

1. Introduction

CNN based methods have recently led to significant advances in the literature of hand pose estimation. Many works, however, are hindered [28, 61, 9, 7, 29] due to limited real datasets [46, 59, 54, 58], and thus rely on synthetically generated data. 3D hand pose estimation from monocular RGB images and video is in particular challenging and has only recently been explored [62].

We need new network architectures, and new real ground truth (GT) datasets to tackle this highly ambigious problem. While the former is easier to achieve and also compare to, unfortunately on very limited monocular datasets captured [59], the latter is quite hard to obtain, and based on the hunger of CNN-s for real data, it seems to also explain the bottleneck behind limited accuracy of various architectures on such monocular RGB based tasks, as opposed to their depth counterparts.

In this work, we propose new learning architectures and high quality datasets to improve the accuracy of 3D hand pose estimation from a single RGB image. Our squeezenet [15] based architecture attempts to map a single RGB hand image directly to a 3D hand representation (using angle differences from a reference neutral pose, similar to [61]), without the necessity to lift from 2D to 3D as in previous works [62, 60, 54]. It is trained on our new, large, realistically rendered hand dataset, consisting of around 3 Million RGB images with respective 3D annotations. By construction, such a model allows to refine itself on real-data in a semi-supervised fashion, showing improved performance on gesture classification tasks (see Sec.5).

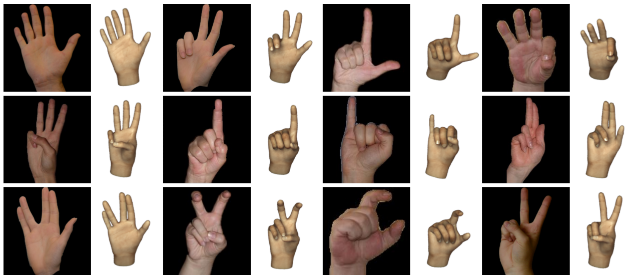

Figure 1. Real predictions on the HGR dataset [16, 27, 13].

A crucial part of our technique is refining the network in an unsupervised way on real unseen monocular data, given that a depth image is provided or extracted. We demonstrate through various experiments that we can obtain a performance boost as compared to training with purely synthetic or limited monocular ground truth data, unlocking further applications that work with RGB monocular data.

We show increased performance as compared to previous works based on monocular RGB images on a variety of tasks (3D pose estimation, hand gesture recognition and 2D fingertip detection), while being on par with methods that require depth as input. Our technique can also be seen as an economic and automatic way of creating a ground truth labeled dataset and we believe will be instrumental in creating new datasets as well.

To summarize, our main contributions are :

- A new realistically rendered hand dataset with 3D annotations available to the community, that helps in hand segmentation and 3D pose inference tasks.
- A method for refining an RGB-based network trained on synthetic data with unlabeled RGB hand images and the corresponding depth maps.

- A state-of-the-art complete system for 3D hand pose estimation and gesture recognition from monocular RGB data that is thoroughly validated on available datasets.

2. Related work

Hand pose estimation methods can be primarily classified with respect to the input as depth, monocular RGB, multi-view, and video-based. Given the low cost of RGBD sensors, there has been a vast amount of work on hand pose estimation based on depth images, which can be further classified as being either generative (model-based), discriminative (appearance based), or both (hybrid) [11]. An additional classification can be made based on how the input is mapped to the output : 2D-to-3D lifting [62, 53, 60, 3, 54, 31] or direct 3D mapping based methods [9, 61, 28]. Our method can be classified as a discriminative, direct 3D mapping method with a monocular RGB as input.

Generative Approaches. Melax *et al.* [22] formulate the hand optimization as a constrained rigid body problem. Schröder *et al.* [37] suggest optimizing in a reduced parameter space and Tagliasacchi *et al.* [45] combine ICP with temporal, collision, silhouette, kinematic and data-driven terms to track with high robustness and accuracy from a depth video. Sharp *et al.* [38] enhance this approach with a smooth model and the possibility of reinitialization. Particle swarm optimization (PSO) approaches have also been used, requiring extensive rendering of an explicit hand model in various poses [30], estimating ground truth [54] for the NYU dataset [54], or combining it with ICP [32] to increase its robustness. Taylor *et al.* [50] minimize an error between a realistically synthesized and real depth image.

Discriminative Approaches. Oberweger *et al.* [28] show how to boost the prediction performance by a projection to a reduced subspace before the final regression, through a bottleneck layer. Zhou *et al.* [61] predict joint rotation angles (similar to us) by proposing a forward kinematic layer, coupled with a physical loss to penalize angles outside a specified range. Similarly, Dibra *et al.* [9] map to angles and show how to refine their CNN on unlabeled depth input images. Ge *et al.* [12] do not make use of depth, but instead project a hand point cloud onto three orthogonal planes and feed the projections into three different CNN-s. Deng *et al.* [7] and Moon *et al.* [25] map 3D volumetric representations though 3D CNNs to the pose in 3D. Apart from CNN-s, there exist also methods that utilize decision forests to make a 3D pose prediction [17, 47, 56]. These methods are typically fully supervised, except for [55, 49] and [9]. We show semi-supervised and unsupervised adaptations, with real RGB and depth data, however applied to RGB input.

Hybrid Approaches. Sometimes, CNN predictions are complemented with an optimization step. Tompson *et al.* [54] first predict hand keypoints and optimize for the actual pose using inverse kinematics. Mueller *et al.* [26] fit the hand skeleton to 2D and 3D joint predictions from a CNN. Ye *et al.* [57] combine CNN-s and PSO in a cascaded and hierarchical manner. Sinha *et al.* [40] first reduce the dimensionality of the depth input through a CNN and then adopt a matrix completion approach with temporal information to optimize for the final pose. Oberweger *et al.* [29] use a deep generative neural network to synthesize depth images, which are utilized to iteratively correct a pose predicted by another network during testing.

Our method can be regarded as an extension and adaptation of data-driven methods that directly map an input to *e.g.* 3D joint angles [61, 4, 48], with the ability to refine themselves in an unsupervised manner to real data [9], being the first to apply this to monocular RGB images instead of depth images as the input.

Video-Based Methods. Since RGBD sensors are not always available, further methods have been proposed, that utilize RGB images in combination with temporal information. La Gorce *et al.* [6] use texture, position and pose information from the previous frame to predict the current pose. Romero *et al.* [35] exploit temporal knowledge to guide a nearest-neighbor search. All these methods have to solve the problem of obtaining a first estimate.

Multi-View-Based Methods. Another approach involves the use of multiple cameras to compensate for the lack of depth data, alleviating the problems with occluded parts. Zhang *et al.* [59] utilize stereo matching for hand tracking, Simon *et al.* [39] apply multi-view bootstrapping for keypoint detection, and Sridhar *et al.* [44] estimate 3D hand pose from multiple RGB cameras, with a hand shape representation based on a sum of Anisotropic Gaussians, whereas [43] combine RGB and Depth data to obtain a richer input space.

Image-Based Methods. Due to the larger availability of regular color cameras, as opposed to the abovementioned methods, we make use of neither depth nor multi-camera or temporal information. One of the first single frame based hand detection works, from Athitsos and Sclaroff [1] utilize edge maps and Chamfer matching. It was only recently that one of the first monocular RGB based methods [62] for 3D hand pose estimation was presented, utilizing CNN-s and synthetic datasets. In contrast to our method, they split the prediction into a 2D joint localization step followed by a 3D up-lifting, and use their own synthetic dataset to complement the scarcity of existing datasets. We utilize our new, high quality, hand synthetic dataset to predict 3D joint angles directly from an RGB image and strongly compare to [62] on various tasks in Sec. 5. Concurrent to our work, there exist methods based on Variational Autoencoders [41] for cross-modal learning and GANs [26] for learning a mapping from synthetic to real hands data, that tackle the same problem of 3D hand pose estimation from RGB images.

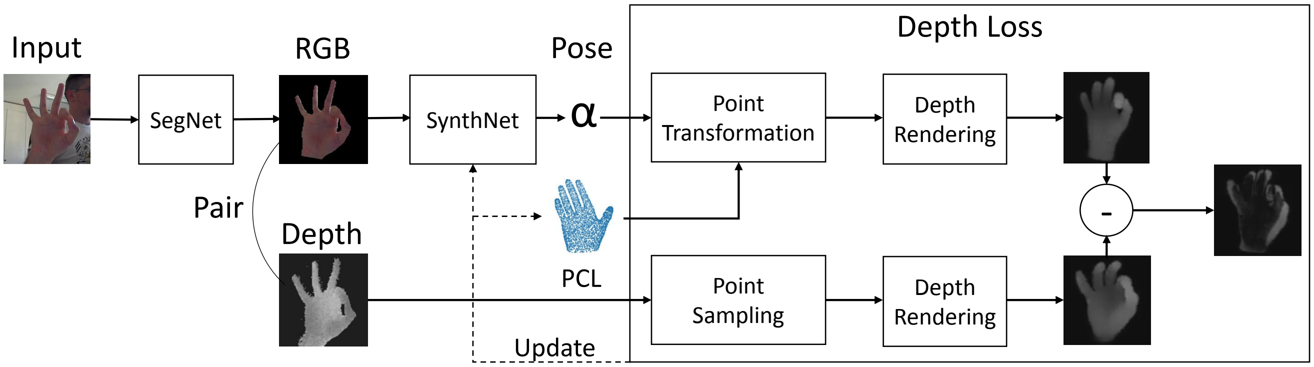

Figure 2. Overview of the training pipeline. Given a monocular RGB image as input, a $SegNet$ [2] based network first segments out the background, the result of which is input into $SynthNet$, a CNN model trained purely on synthetic data (Sec.4) that predicts the hand pose in terms of angles α. In order to fine-tune the network to real monocular data, provided that a corresponding depth image is given, we augment the initial base network with a depth loss component. We refer to this combination during training time as $RefNet$. Given α as well as PCL, a point cloud that initially represents our hand model and gets iteratively updated to the input one, the weights of $SynthNet$ can be updated without the need of labeled data. At test time, a forward pass through $SegNet$ and $SynthNet$ estimates the desired pose.

3. 3D Hand Pose Estimation and Refinement

The overview of our method is depicted in Fig. 2. We attempt to achieve two main goals : 1. estimate the 3D hand pose, given a single monocular RGB image, and 2. enable a refinement of our method predictions on unseen real images in an unsupervised way. Due to the lack of real RGB ground truth datasets, we tackle the first goal, by training a CNN ($SynthNet$ Sec.3.2) that minimizes an angle loss ($\mathcal{L}^{angle}$) in a supervised manner. We train purely on a new large synthetic dataset (Sec.4), consisting of masked-out renderings of hands in various poses, shapes, illuminations and textures and their respective 3D annotations. At test time, we first segment a raw RGB image in order to obtain only the hand part, by passing it through a segmentation CNN (Sec.3.1), trained on a combination of real and our own synthetic data to minimize a categorical cross-entropy loss ($\mathcal{L}^{mask}$). This first part captures priors on the variability of possible free hand poses already at training time and achieves results on-par or even better than state-of-the-art works on real datasets for a variety of tasks (Sec.5).

We tackle the second goal of real data based refinement, by extending our $SynthNet$ with a component based on a depth loss ($\mathcal{L}^{depth}$), which allows it to get fine-tuned on unseen unlabeled real RGB data, provided that an analogue unlabeled depth image (registered or unregistered), is present at training time. We refer to this combination during training time as $RefNet$, which can be considered as a differentiable renderer. The weights of $SynthNet$ are adapted to real data in an unsupervised manner. During test time, a forward pass through it allows to estimate the 3D pose. This second part is very important, because of the known discrepancy between real and synthetic data due to different hand shapes, poses, sensors, and environment conditions. This refinement leads to significant improvements over the network trained purely on synthetic data, which we show through experiments in Sec.5 and supplementary.

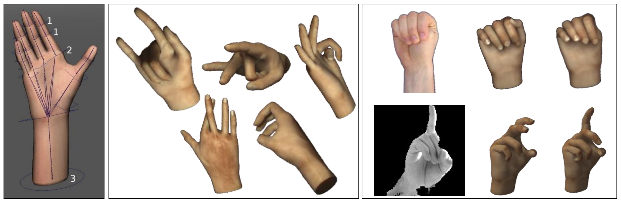

Figure 3. (Left) Rigged hand model with max 3 (rotational) × 17 (joints) DOF (Middle) 5 samples from our dataset in 5 different orientations (Right) Two semi-supervised refinement examples from our own dataset (top) and $Senz3D$ [23, 24] (bottom) - from left to right : input, $SynthNet$ unrefined and refined prediction.

3.1. Hand Segmentation Nets

Before segmentation, the hand needs to be localized in the image. Inspired by He et al. [14] that compute object detection and segmentation operating in two stages with Faster R-CNN [34], we adopted SegNet [2] to first propose the hand region and then compute a pixel-wise mask of the hand. The detection is also performed via segmentation, producing a rough mask to localize the hand and crop around it, which in turn is utilized to produce a more refined hand mask. In order to decrease training and inference time, without affecting accuracy, we removed some layers from both the encoder (two convolutions and one max-pooling) and decoder (8 convolutions and one up-sampling). We call this architecture *OurSegNet* and provide details in the supplementary. Segmentation is a necessary preprocessing step of our pipeline, and not a contribution of this work, hence throughout this paper we analyze both it‘s performance

(Sec.5) and that of *HandSegNet* from [62]. The expected input RGB image and segmented output are 256×256 pixels each. The latter serves as input for the next stage.

3.2. Synthetic RGB CNN Model (SynthNet)

Inspired by recent work [5] that trains a SqueezeNet [15] based architecture purely on realistically rendered masked-out synthetic garment images to map directly to 3D garment vertex meshes, we also pose our problem as finding a mapping from masked-out images of hands to the 3D hand pose. We start by training a SqueezeNet model (*SynthNet*) adapted to regression (details in the supplementary) purely on our synthetically generated dataset (Sec.4), which directly predicts, as in [8, 5, 61, 21], a 3D pose α from a (masked-out) RGB image I (Sec.5). Our 3D pose α is represented in euler angles, similar to Zhou *et al*. [61], however quaternions or rotation matrices can be utilized too. The main constraint is that α must be informative enough to calculate a forward kinematic chain, yielding the exact information on how each joint transforms to the predicted pose (Sec. 3.3). This is made possible by our rigged hand model (Fig.3 (Left)). More specifically, α is given as an angle difference for each of the hand joints from the joint angles of a hand in a neutral pose (open palm). Given the RGB images of the synthetic training data, we train our SynthNet from scratch to minimize the mean squared error ($\mathcal{L}^{angle}$) between the pose from our dataset and the predicted pose. We noticed that by first converting the input images to grayscale and then applying histogram normalization, with one and 99 percentile as borders to remove pixel outliers, not only made the network converge faster, but also helped with skin-color invariance. Since during training, all the hand masks are centered, at test time we also center and scale the hands to a square image of 225×225 pixels (similar to SqueezeNet input), when necessary padded at the borders.

Semi-Supervised Refinement on Real RGB Images. One advantage of utilizing angles instead of joint positions, is that they can be easily restricted to the allowed Degrees of Freedom, reducing the large space of infeasible poses, and constraining the latent space [4]. Given a skeleton, angles can easily be converted to joints and hence fully determine a pose. This might penalize accuracy on exact 3D joint estimation tasks, under fixed hand skeleton model assumptions, however it can be quite attractive for other tasks where the hand skeleton constellation is more important than the exact joint position, e.g. hand gesture recognition/classification. Another advantage of utilizing angles, is that it allows any pre-trained fully supervised network (regardless whether real or synthetic data is used), to refine itself on easily obtainable real unlabeled RGB images. Real images of hands in various shapes, skin colors, lighting conditions and rotations can be easily captured with cheap RGB sensors, under the constraint that users perform pre-specified gestures, as in [23, 24]. These gestures can be easily modeled, given a synthetic hand model, obtaining the ground truth (angles) without additional manual effort. Angles are advantageous here, as various user poses would map to the same ground-truth, regardless of the exact hand position and rotation in the image. In this way, the input space is enriched with multiple real images that map to the same angles, which in turn helps to fine-tune synthetic networks and improve the gesture recognition predictions (Fig.3 (right) and Sec.5).

3.3. Unsupervised Refinement from Depth Images

$SynthNet$ alone gives good initial predictions on various real data ((Sec.5), Fig.1 and Fig.5), however a discrepancy between synthetic and real datasets is known in literature. Inspired by works based on differentiable renderers [20] and differentiable offline [33, 9] and online [52] depth based calibration and refinement, we extend our network with a component that enables $SynthNet$ to get refined unsupervised, trained to minimize a depth loss ($\mathcal{L}^{depth}$) on unlabelled depth data, that have one-to-one correspondences to the input real RGB images. Let‘s assume we have pairs of RGB and Depth images (I, D). Acquisition of such pairs is very cheap with today‘s RGBD sensors (Sec.5). Based on the approach from [9], we compare the input depth image D to a synthesized depth image D_I, which is computed from $SynthNet$ predicted pose α, given I as input, and a pointcloud PCL sampled from the hand mesh model, in order to predict the accuracy on unlabelled data. We transform the PCL points according to α, applying Linear Blend Skinning (LBS) [19], and subsequently render them to obtain a synthetic depth image.

Pointcloud Transformation. Similar to [61], we compute the forward kinematic chain, which yields for each joint the transformation matrix, transforming from the model space of a neutral pose into the model space of the skinned pose α. What is important is that this step is differentiable, since only matrix multiplications and trigonometric functions are required. We denote with $T(\alpha) = [T_1(\alpha_1), \ldots, T_J(\alpha_J)]$ these transformation matrices, where J is the number of joints used (see supplementary for details). In contrast to [61], we do not just transform the joint positions, but a bigger set of points $PCL = [p_1, \ldots, p_n]$ representing the whole hand. Each point p_i is associated with one or more joints. The weight $w_{i,j}$ defines how much the point is bound to the joint j. Applying LBS [19] $f^{LBS}(PCL, T)$ transforms each point by a linear combination of the matrices T_j according to its weights:

$$\hat{p}_i := f_i^{LBS}(PCL, T) = \sum_{j=1}^{J} w_{i,j} T_j(\alpha_j) p_i \qquad (1)$$

Because this formulation is not just differentiable with respect to T (a very important property that allows backpropagation to the $SynthNet$ model), but also with respect to

PCL, we can relax the static hand model to a dynamic one, that gets updated during training to automatically adapt to the hand shape. In order to give an intuitive advantage of this approach, imagine a personalized adaption to a different real person‘s hand shape, starting from a non-parametric 3D hand model. This becomes important since in reality, not only the poses change but also the hand shapes.

Depth Rendering. The 3D hand shape and pose can be adapted to the real hand shape and pose by iteratively minimizing a difference in depth projections ($\mathcal{L}^{depth}$), of points PCL and PCL_D, sampled from the hand model and the input depth image D, respectively. PCL is uniformly sampled from the hand mesh. In order to render PCL in a differentiable way, we select only the points with the lowest z-value (closest to the camera) for each of the image coordinates ($D_{i,j}$), and weight the z-value of each point with a 2D basis function ϕ around its position. This weighting (smoothing) step is important since otherwise, only picking a depth value at each widely spaced sampled point would make the method non-differentiable. Let $p_i = [p_{i,x}, p_{i,y}, p_{i,z}] \in PCL$. The rendered depth image approximation is defined as:

$$D_{I\,i,j} := f_{i,j}^{\text{depth}}(PCL) = \max_k(\text{depth}_{i,j}(p_k)) \qquad (2)$$

where the points are assumed to be in the $[0,1]$ range and the z-values represent the depth values with respect to the camera:

$$\text{depth}_{i,j}(p) = (1 - p_z)\phi_{i,j}(p) \qquad (3)$$

Let $\text{dist}^2_{i,j}(p) = (j - p_x)^2 + (i - p_y)^2$. $\phi \in C^1$ was chosen to have finite spatial support of a circle with radius r, and can be defined as:

$$\phi_{i,j}(p) = \left(1 - \left(\frac{\text{dist}_{i,j}(p)}{r}\right)^2\right)^2 \mathbb{1}_{\text{dist}^2_{i,j}(p) < r^2} \qquad (4)$$

Due to the discrepancy between the synthesized (D_I) and real depth (D) images, as also motivated in [9], we do not directly compute the loss, but instead also sample a point cloud PCL_D from the real depth image D and render it using f^{depth} to obtain D_S. The actual loss taken in the end is the $L1$ norm of the difference between both synthesized images, Fig.2 :

$$\mathcal{L}^{\text{depth}} = \sum_{i,j} |f_{i,j}^{\text{depth}}(f^{LBS}(PCL, T(\alpha))) - f_{i,j}^{\text{depth}}(PCL_D)| \qquad (5)$$

4. Synthetic Dataset Generation

In the absence of monocular RGB labelled datasets, in order to capture the space of pose variability already at training time, we create a new, large, realistically rendered, available free-hand dataset.

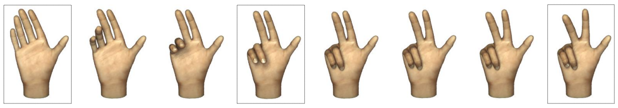

Figure 4. Three base poses (in boxes) with linear interpolation on the parameter space in between.

Hand Model. We opted for a commercial rigged and textured hand model[1] for Maya®[2]. The skeleton consists of 21 bones with 51 degrees of freedom (DoF), see Fig.3 (Left). Since not all the DoF are feasible for a human skeleton, we restrict our method to 4 DoF per finger and 3 for the rotation of the wrist. A real human hand has more than these 23 DoF [18], however, the additional DoF are often ignored to simplify the problem [45].

Synthetic Dataset. Inspired by [56], we decided to use a combination of manual and automatic sampling. We first create some base poses. Then we linearly interpolate over the parameters between each pair of base poses to generate new poses, as in Fig.4, detecting intersections. This procedure allows to easily adapt the dataset to a desired purpose by crafting suitable base poses and then automatically generating the linear span between them. We end up with 399 such poses. Details on the proposed enumeration of base poses for a general purpose system can be found in the supplementary material, along with a heat map visualization experiment demonstrating our network's capability to be trained on the generated poses (with minor difficulty on typically occluded parts, such as the thumb). In addition to the varying poses, for each view (we consider 5 views - front, back, both sides and top, Fig.3 (Middle)) we apply 5 random rotations (45 degrees for each DoF of the wrist joint) and illumination changes to each image. We also vary the texture and shape.

Collision Avoidance. Since a linear interpolation within the hand pose space can lead to self-intersection, the automatic generation of new poses contains an intersection detection which rejects such undesired poses. In order to detect intersections, we loop over all finger vertices to find the nearest (other) finger neighbor. By projecting the vertices difference vector onto the other finger surface normal, it can be computed whether the vertex is inside the foreign mesh or not. An intersection occurrence is detected when an “inside” threshold is passed. In order to simulate flesh interaction between fingers, we relax the threshold allowing very little intersection. Due to interpolation with collision avoidance we end up with 122106 different poses.

Un-natural Poses. The linear interpolation preserves many constraints applied to the base poses, *e.g.* maximal angle-range and fixed ratio between certain angles. Thus, it suf-

[1] https://www.turbosquid.com/3d-models/rigged-male-hand-max/786338

[2] www.autodesk.com/products/maya

fices to create the base poses with the desired constraints to make sure that the same holds within the complete dataset.

5. Experiments and Results

5.1. Training and Test Datasets

Detection and Segmentation Datasets. We utilize the method and dataset from [62], for hand bounding box detection. On the other hand, for segmentation we use both real and synthetic data. The real hand dataset contains 19000 images, 6000 of which come from the Hand Gesture Recognition (HGR) dataset [16, 27, 13], which is an augmentation of the initial 1500 raw images (consisting of 33 individuals and 70 gestures), that we segment, add various backgrounds and perform in-plane rotations of the hand. The remaining 13000 belong to three individuals, captured performing various poses in front of a green screen, which is replaced with a random background. The synthetic images are in the $100K$ range and come from our synthetic dataset.

Pose Inference Datasets. Many publicly available datasets are shot with depth cameras, *e.g.* the recently introduced BigHand2.2M Dataset [58]. There is a lack of proper RGB datasets. The NYU Hand Pose Dataset [54] *e.g.* contains holes in the RGB images if no depth data is available, while the Dexter RGBD dataset [42] has incomplete hand annotation (fingertips) [62]. We make use of the Stereo Hand Tracking Dataset [59] (StereoDS), which contains twelve motion sequences in front of various backgrounds (B1 through B6, and for each set, a count and random sequence of 1500 images each), which provides RGB and Depth images together with the 3D joint positions. Another area having a rich variety of RGB datasets is hand gesture recognition, where the ground truth is a class label. We utilize the German Fingerspelling Database (RWTH) [10], that provides the classes of 35 gestures from the German sign language, for 20 people, HGR [16, 27, 13], which in addition to the class provides visible 2D fingertip locations and Senz3D [23, 24] containing 11 gestures performed by 4 different people repeated 30 times each. Additionally, to demonstrate unsupervised refinement on real data, we capture our own dataset (*IntelDS*) utilizing the Intel RealSense Camera. It consists of 1000 pairs of registered RGB and depth images for testing and $30,000$ for training (in the size of 120×120 pixels and without GT annotations), from one individual wearing a black wristband, that allows for a simple intensity based segmentation.

5.2. Segmentation Accuracy Improvement

We evaluate the segmentation accuracy for both *HandSegNet* [62] and *OurSegNet*, when training is performed with and without adding our synthetic dataset to the available real ones. We evaluate on B1 random and count (150 images each) of $StereoDS$ and the complete $RWTH$, observing an accuracy increase in the latter case (Table 1).

Dataset	*[62]*	*[62]+Synth*	*OSN*	*OSN+Synth*
B1 Random	91.5	97.7	91	95.5
B1 Count	92	98	92	96
RWTH	93.34	93.37	92.9	93,1

Table 1. Segmentation accuracy in % for *HandSegNet* [62] and *OurSegNet* (OSN) trained with and without our synthetic dataset.

5.3. Refinement with Unlabeled Data

Semi-Supervised on Real RGB Images. As a proof-of-concept, we utilize the $Senz3D$ dataset [23, 24], to fine-tune our $SynthNet$ on real RGB images, by splitting the dataset in half (300 each) for training and testing for a gesture classification task on 10 of the classes. We first manually craft a synthetic pose for each of the classes, in order to obtain approximate GT labels (angles) for each training image. Then, we learn a mapping from angles to classes, similar to [62]. We measure the accuracy utilizing a 10-fold cross validation, and notice an increase from 94 to 96.7%, which is enabled by representing the 3D pose in terms of angles as opposed to 3D joints (Sec.3.2). Fig.3 (Right) visualizes this improvement along with the supplementary video.

Unsupervised on Pairs of RGB and Depth Images. We utilize the $IntelDS$ to refine our $RefNet$ in an unsupervised way, utilizing pairs of RGB and depth data, and compare it to the results of $SynthNet$ before refinement. We visualize the results before and after refinement in Fig.6, also through videos and ROC curves in the supplementary, demonstrating a clear improvement after the refinement. By computing MSE between the two synthesized images which are utilized to compute the depth loss, we notice that the error halves in the latter case. Additionally we show a video where we compare to [9] trained on depth images and demonstrate similar performance.

5.4. Comparison to State-of-the-art

We compare to related methods working on RGB or depth input images, and investigate generalization on various dataset, for three main tasks : gesture recognition, 2D fingertip estimation and 3D pose estimation. Qualitative results on predictions are depicted in Fig.1 and Fig.5 as well as in the supplementary material.

Classification on Spelling Dataset. Like Zimmermann *et al.* [62], we evaluate our system on $RWTH$ on all the 30 static gestures, by first predicting the poses and then applying a pose classifier to the respective class. Unlike [62], we do not utilize images from this dataset to refine on and we first segment the images utilizing $OurSegNet$. We utilize 10-fold cross validation to estimate the accuracy since no split specification was given by [62]. Training was done with one hidden layer of 500 neurons with Relu activation and dropout probability of 0.5. We achieve superior perfor-

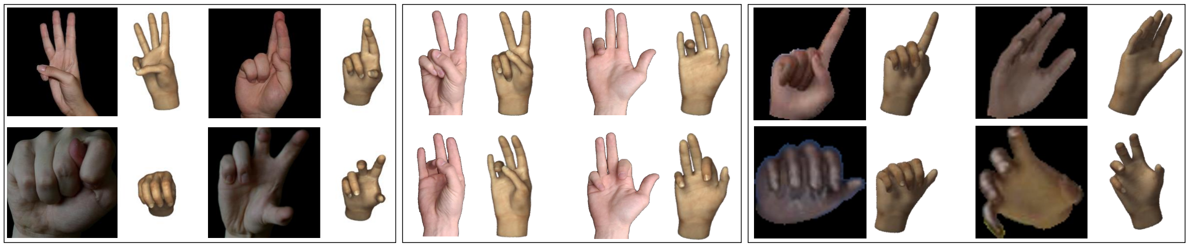

Figure 5. $SynthNet$ predictions on (left) HGR dataset [16, 27, 13] (middle) one individual hand (right) synthetic dataset from [62].

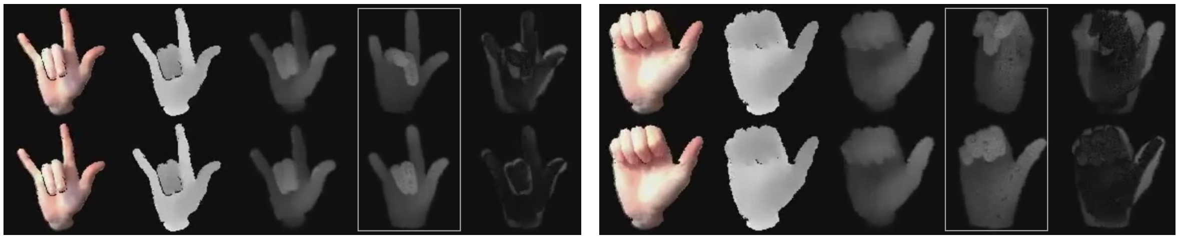

Figure 6. Two examples from our validation set $IntelDS$. $SynthNet$ predictions before (top) and after refinement (bottom). From left to right : RGB Input (I), Input Depth (D) , Synthesized Input Depth (D_S), Prediction (D_I) and Error in depth prediction.

Method	*RWTH*	*Senz3D*
[10] on subset (from [62])	63.44	-
[62]	66.8	77
Ours	**73.6**	**94**

Table 2. Classification accuracy comparison, in % of correctly classified poses, on the RWTH [10] and Senz3D [23, 24].

Method	*Error*
[62] (their segmentation)	804.23 px^2
[62] (oracle segmentation)	483.28 px^2
Ours (oracle segmentation)	**361.47** px^2

Table 3. Fingertip accuracy on the HGR Dataset [16, 27, 13] computed as MSE over pixel errors, with image size 225 × 225 pixels.

Evaluated for \Trained on	*Joint positions*	*Joint angles*
Joint Position MSE	**0.199**	0.397
Joint Angle MSE (deg)	42.829	**12.763**

Table 4. Joint Angles vs Positions MSE on our synthetic dataset.

mance compared to [62] and [10] as shown in Table 2. We repeat the same experiment, however now on $Senz3D$ over 10 classes, also achieving a better performance than [62].

Fingertip Detection Comparison. We evaluate $SynthNet$ predictions on the HGR dataset, which contains hands from multiple people, assuming an oracle segmentation (ground truth segmented by us). Fig.1 and Fig.5 (left) shows a qualitative assessment of our results, where the predicted pose seems quite accurate, despite training only on synthetic data. To quantitatively compare to [62], we measure the accuracy of predicting 2D (visible) joint positions, by computing the MSE on pixels for all front facing images (since back facing ones have almost no visible fingertip). Zimmermann *et al.* [62] provide 3D joints directly, while we apply the kinematic chain on angles α to retrieve the 3D joints. These 3D fingertips are then projected into 2D by solving a least-squares system to best fit to the groundtruth labels (since no camera info is given). Table 3 depicts these results, with [62] evaluated with their and the oracle segmentation (since we train $OurSegNet$ on HGR we only evaluate on oracle segmentation), where our method achieves higher accuracy.

ROC Angle and 3D Joint Curves. We evaluate accuracy on 3D pose prediction for different methods by computing ROC curves, that denote the fraction of frames below a maximum 3D joint (or angle) prediction error, on the B1 set of $StereoDS$. We compare to [62], that assume an RGB input as we do, and four other depth-based methods. Such methods are trained to directly predict 3D joint positions, unlike ours that predicts angles (Sec.3.2), and hence minimizes a different quantity (*e.g.* a slight wrist angle misscalculation would bring a larger error on 3D joints prediction, even if the rest of the angles are correctly predicted). Thus, we argue that a direct comparison on this dataset is not possible, also due to the discrepancy between the GT skeleton in $StereoDS$ and our hand model skeleton, from which we compute 3D joints from angles. In order to back this up, we performed an experiment, on 300 unseen samples from our synthetic dataset, where we once trained for 3D joint positions and once for angles, and computed the MSE for both cases. As it can be noticed in Table 4, training for the respective task always achieves a smaller error. Nevertheless, for completeness we compare on this dataset and

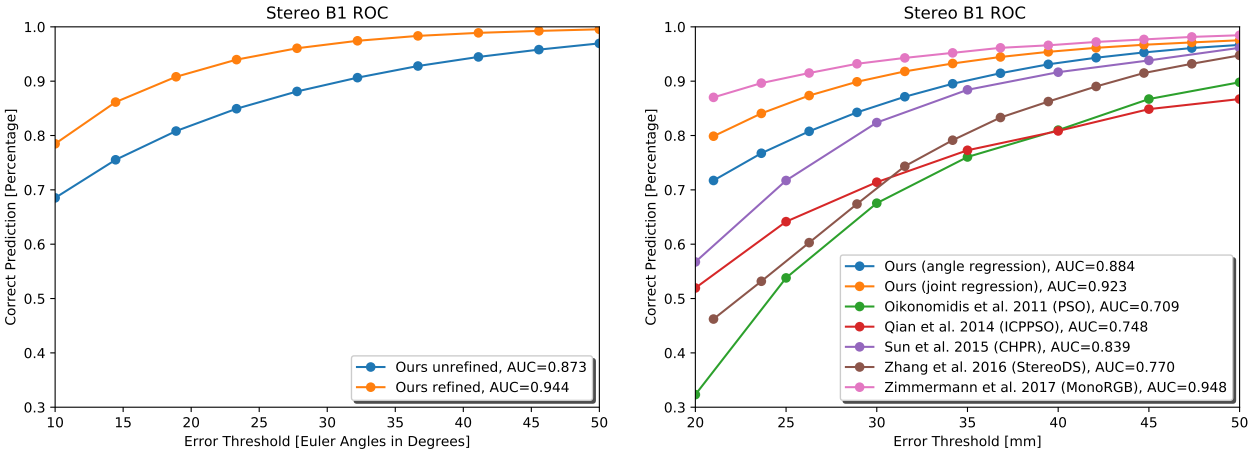

Figure 7. Accuracy on the $StereoDS$ [59] dataset. (Left) Improvement in euler angles due to refinement (Right) Comparison to state-of-the-art methods trained to map onto 3D joints. We show our ROC curve trained on angles along with a version trained on joints.

report ROC curves for both angles and joints, in Fig.7. Due to the lack of GT segmentation we first apply $OurSegNet$ to obtain the masked-out RGB images. The methods we compare to, refine on sets B2-B6 consisting of $15,000$ images. We can not directly fine-tune on such datasets unfortunately, however we apply the following procedure : we compute the GT angles over B3-B5 (note from a different skeleton) and utilize this as our GT for refinement on the training set. Due to innacurate segmentation we do not make use of B2 and B6. We then apply forward kinematics to obtain the 3D joints from angles, and learn a linear mapping from our skeleton predicted 3D joints to those of the $StereoDS$ GT, in order to minimize the bias between both skeletons. At test time, we first predict the angles on B1, then compute joints and apply the mapping. The results are depicted in Fig.7 (Right) with [62] achieving (as expected) a higher Area Under Curve (AUC). Nevertheless, computing the ROC for euler angle errors, as in Fig.7 (Left), we notice that the AUC for our method after refinement is almost the same as that of [62]. In order to quantitatively prove our claim for the discrepancy between training for different tasks, we additionally train a network to predict 3D joints instead of angles, utilizing only our synthetic data and refining on B3-B5. We already notice a boost in the predictions, with the new curve, Fig.7 ((Right) Ours (joint regression)), reaching similar accuracy to that of [62]. We think that the difference between the curves can be due to our refinement only on a part of the complete training set that [62] was refined on.

6. Discussion and Conclusions

We could show, through quantitative and qualitative evaluations, that utilizing lightweight CNN-s trained purely on our newly proposed synthetic dataset can achieve accurate pose inference, for a variety of tasks, strongly competing with and even outperforming existing state-of-the-art. We additionally showed that by extending its construction with a depth loss component, coupled with our pose representation, the accuracy further improved via semi-supervised and unsupervised training with real unlabeled images.

At the moment, we utilize training data generated from a single shape hand model. Despite the fact that we could show generalization on multiple real hands, and good accuracy especially on classification tasks, there is still room for improvement, *e.g*. experimenting with adding a second shape improved prediction on HGR by 10% (supplementary). Additionally, due to the joint angle parametrization, the same parameters could represent different poses when children and adult hands are considered. Our current optimization model, though, allows an internal adaptation to a hand shape. Coupling our method with recent more powerful hand shape models such as Tkach *et al*. [51] and Romero *et al*. [36]'s has the potential to improve and personalize hand pose estimation for a variety of human hand shapes.

Even though we could show improvements in segmentation, based on the synthetic dataset, most of it is due to the real GT training data we annotated. As also backed up by our refinement experiments, further real GT datasets with segmentation and pose annotations are very important. Additionally, we could avoid segmentation, by synthesizing 3D models in front of various backgrounds, however on the expense of added training time and larger datasets.

Lastly, we envisage to apply our technique to related tasks such as human pose estimation, with minimal changes to the underlying representation and architecture.

Acknowledgement. We thank Niko Benjamin Huber and the reviewers for their comments on the paper.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 432–442, 2003. 2

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. 3

[3] F. Bogo, A. Kanazawa, C. Lassner, P. V. Gehler, J. Romero, and M. J. Black. Keep it SMPL: automatic estimation of 3d human pose and shape from a single image. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 561–578, 2016. 2

[4] C. Choi, S. Kim, and K. Ramani. Learning hand articulations by hallucinating heat distribution. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3123–3132, 2017. 2, 4

[5] R. Danecek, E. Dibra, A. C. Öztireli, R. Ziegler, and M. H. Gross. Deepgarment : 3d garment shape estimation from a single image. *Comput. Graph. Forum*, 36(2):269–280, 2017. 4

[6] M. de La Gorce, D. J. Fleet, and N. Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1793–1805, Sept 2011. 2

[7] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang. Hand3d: Hand pose estimation using 3d neural network. *CoRR*, abs/1704.02224, 2017. 1, 2

[8] E. Dibra, H. Jain, A. C. Öztireli, R. Ziegler, and M. H. Gross. Human shape from silhouettes using generative hks descriptors and cross-modal neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, July 21-26, 2017*, 2017. 4

[9] E. Dibra, T. Wolf, A. C. Öztireli, and M. H. Gross. How to refine 3d hand pose estimation from unlabelled depth data ? In *Fifth International Conference on 3D Vision, 3DV 2017, Qingdao, China, 2017*, 2017. 1, 2, 4, 5, 6

[10] P. Dreuw, T. Deselaers, D. Keysers, and H. Ney. Modeling image variability in appearance-based gesture recognition. In *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, pages 7–18, Graz, Austria, May 2006. 6, 7

[11] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Comput. Vis. Image Underst.*, 108(1-2):52–73, Oct. 2007. 2

[12] 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 *Proc. CVPR*, 2016. 2

[13] T. Grzejszczak, M. Kawulok, and A. Galuszka. Hand landmarks detection and localization in color images. *Multimedia Tools and Applications*, 75(23):16363–16387, 2016. 1, 6, 7

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017. 3

[15] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016. 1, 4

[16] M. Kawulok, J. Kawulok, J. Nalepa, and B. Smolka. Self-adaptive algorithm for segmenting skin regions. *EURASIP Journal on Advances in Signal Processing*, 2014(170):1–22, 2014. 1, 6, 7

[17] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. *Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests*, pages 852–863. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 2

[18] J. Lee and T. L. Kunii. Model-based analysis of hand posture. *IEEE Comput. Graph. Appl.*, 15(5):77–86, Sept. 1995. 5

[19] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 4

[20] M. M. Loper and M. J. Black. OpenDR: An approximate differentiable renderer. In *Computer Vision – ECCV 2014*, volume 8695 of *Lecture Notes in Computer Science*, pages 154–169. Springer International Publishing, Sept. 2014. 4

[21] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation using transfer learning and improved CNN supervision. *CoRR*, abs/1611.09813, 2016. 4

[22] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, GI '13, pages 63–70, Toronto, Ont., Canada, Canada, 2013. Canadian Information Processing Society. 2

[23] A. Memo, L. Minto, and P. Zanuttigh. Exploiting silhouette descriptors and synthetic data for hand gesture recognition. In *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference, Verona, Italy, October 15-16 2015.*, pages 15–23, 2015. 3, 4, 6, 7

[24] A. Memo and P. Zanuttigh. Head-mounted gesture controlled interface for human-computer interaction. In *Multimedia Tools and Applications*, 2017. 3, 4, 6, 7

[25] G. Moon, J. Y. Chang, and K. M. Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. *CoRR*, abs/1711.07399, 2017. 2

[26] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Ganerated hands for real-time 3d hand tracking from monocular RGB. *CoRR*, abs/1712.01057, 2017. 2

[27] J. Nalepa and M. Kawulok. Fast and accurate hand shape classification. In S. Kozielski, D. Mrozek, P. Kasprowski, B. Malysiak-Mrozek, and D. Kostrzewa, editors, *Beyond Databases, Architectures, and Structures*, volume 424 of *Communications in Computer and Information Science*, pages 364–373. Springer, 2014. 1, 6, 7

[28] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *CoRR*, abs/1502.06807, 2015. 1, 2

[29] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 3316–3324, Washington, DC, USA, 2015. IEEE Computer Society. 1, 2

[30] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pages 101.1–101.11, 2011. 2

[31] P. Panteleris, I. Oikonomidis, and A. A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. *CoRR*, abs/1712.03866, 2017. 2

[32] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, 2014. 2

[33] E. Remelli, A. Tkach, A. Tagliasacchi, and M. Pauly. Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In *Proceedings of the International Conference on Computer Vision*, 2017. 4

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[35] J. Romero, H. Kjellstrm, and D. Kragic. Monocular real-time 3d articulated hand pose estimation. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 87–92, Dec 2009. 2

[36] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. (*) Two first authors contributed equally. 8

[37] M. Schröder, J. Maycock, H. Ritter, and M. Botsch. Real-time hand tracking using synergistic inverse kinematics. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 2

[38] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. 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. 2

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

[40] A. Sinha, C. Choi, and K. Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[41] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 2

[42] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time joint tracking of a hand manipulating an object from RGB-D input. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 294–310, 2016. 6

[43] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 2456–2463, Washington, DC, USA, 2013. IEEE Computer Society. 2

[44] S. Sridhar, H. Rhodin, H.-P. Seidel, A. Oulasvirta, and C. Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *Proceedings of the International Conference on 3D Vision (3DV)*, Dec. 2014. 2

[45] A. Tagliasacchi, M. Schröder, 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. 2, 5

[46] D. Tang, H. J. Chang, A. Tejani, and T. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 3786–3793, 2014. 1

[47] D. Tang, H. J. Chang, A. Tejani, and T. Kim. Latent regression forest: Structured estimation of 3d hand poses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7):1374–1387, 2017. 2

[48] D. Tang, J. Taylor, P. Kohli, C. Keskin, T. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3325–3333, 2015. 2

[49] D. Tang, T. H. Yu, and T. K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *2013 IEEE International Conference on Computer Vision*, pages 3224–3231, Dec 2013. 2

[50] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Trans. Graph.*, 35(4):143:1–143:12, July 2016. 2

[51] A. Tkach, M. Pauly, and A. Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)*, 2016. 8

[52] A. Tkach, A. Tagliasacchi, E. Remelli, M. Pauly, and A. Fitzgibbon. Online generative model personalization for hand tracking. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)*, 2017. 4

[53] D. Tomè, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CoRR*, abs/1701.00295, 2017. 2

[54] 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, 6

[55] C. Wan, T. Probst, L. J. V. Gool, and A. Yao. Crossing nets: Dual generative models with a shared latent space for hand pose estimation. *CoRR*, abs/1702.03431, 2017. 2

[56] C. Xu, A. Nanjappa, X. Zhang, and L. Cheng. Estimate hand poses efficiently from single depth images. *Int. J. Comput. Vision*, 116(1):21–45, Jan. 2016. 2, 5

[57] Q. Ye, S. Yuan, and T. Kim. Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. *CoRR*, abs/1604.03334, 2016. 2

[58] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T. Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. *CoRR*, abs/1704.02612, 2017. 1, 6

[59] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. 3d hand pose tracking and estimation using stereo matching. *CoRR*, abs/1610.07214, 2016. 1, 2, 6, 8

[60] R. Zhao, Y. Wang, and A. M. Martínez. A simple, fast and highly-accurate algorithm to recover 3d shape from 2d landmarks on a single image. *CoRR*, abs/1609.09058, 2016. 1, 2

[61] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. In *IJCAI*, 2016. 1, 2, 4

[62] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single RGB images. *CoRR*, abs/1705.01389, 2017. 1, 2, 4, 6, 7, 8