DeepHPS: End-to-end Estimation of 3D Hand Pose and Shape by Learning from Synthetic Depth

Jameel Malik[1], Ahmed Elhayek[1], Fabrizio Nunnari[2], Kiran Varanasi[1],
Kiarash Tamaddon[2], Alexis Heloir[2], and Didier Stricker[1]

[1]*AV group, DFKI Kaiserslautern, Germany*
[2]*DFKI-MMCI, SLSI group, Saarbruecken, Germany*
{jameel.malik,ahmed.elhayek,fabrizio.nunnari,kiran.varanasi,
kiarash.tamaddon,alexis.heloir,didier.stricker}@dfki.de

Abstract

Articulated hand pose and shape estimation is an important problem for vision-based applications such as augmented reality and animation. In contrast to the existing methods which optimize only for joint positions, we propose a fully supervised deep network which learns to jointly estimate a full 3D hand mesh representation and pose from a single depth image. To this end, a CNN architecture is employed to estimate parametric representations i.e. hand pose, bone scales and complex shape parameters. Then, a novel hand pose and shape layer, embedded inside our deep framework, produces 3D joint positions and hand mesh. Lack of sufficient training data with varying hand shapes limits the generalized performance of learning based methods. Also, manually annotating real data is suboptimal. Therefore, we present SynHand5M: a million-scale synthetic dataset with accurate joint annotations, segmentation masks and mesh files of depth maps. Among model based learning (hybrid) methods, we show improved results on our dataset and two of the public benchmarks i.e. NYU and ICVL. Also, by employing a joint training strategy with real and synthetic data, we recover 3D hand mesh and pose from real images in 3.7ms.

1. Introduction

3D hand pose estimation is essential for many computer vision applications such as activity recognition, human-computer interaction and modeling user intent. However, the advent of virtual and augmented reality technologies makes it necessary to reconstruct the 3D hand surface together with the pose. Recent years have seen a great progress in the pose estimation task primarily due to significant developments in deep learning and the availability of low cost commodity depth sensors. However, the stated problem is still far from being solved due to many challenging factors that include large variations in hand shapes, view point changes, many degrees of freedom (DoFs), constrained parameter space, self similarity and occlusions.

Large amounts of training data, enriched with all possible variations in each of the challenging aspects stated above, are a key requirement for deep learning based methods to generalize well and achieve significant gains in accuracy. The recent real dataset [53] gathers a sufficient number of annotated images. However, it is very limited in hand shape variation (i.e. only 10 subjects). Progress in essential tasks such as estimation of hand surface and hand-part segmentation is hampered, as manual supervision for such problems at large scale is extremely expensive. In this paper, we generate a synthetic dataset that addresses these problems. It not only allows us to create virtually infinite training data, with large variations in shapes and view-points, but it also produces annotations that are highly accurate even in the case of occlusions. One weakness of synthetic datasets is their limited realism. A solution to this problem has been proposed by [32, 18], where a generative adversarial training network is employed to improve the realism of synthetic images. However, producing realistic images is not the same problem as improving the recognition rates of a convolutional neural network (CNN) model. In this paper, we address this latter problem, and specifically focus on a wide variation of hand shapes, including extreme shapes that are not very common (in contrast to [30]). We present SynHand5M: a new million scale synthetic dataset

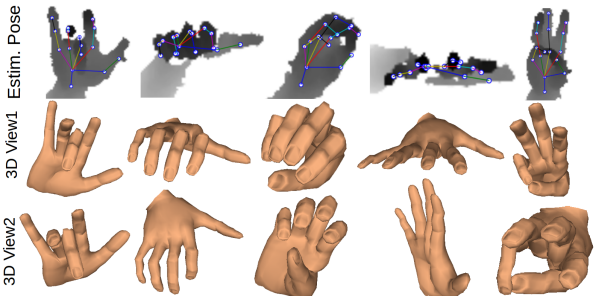


Figure 1: **Real hand pose and shape recovery**: We describe a deep network for recovering the 3D hand pose and shape of NYU[43] depth images by learning from synthetic depth. Note that we infer 3D pose and shape even in cases of missing depth and occluded fingers.

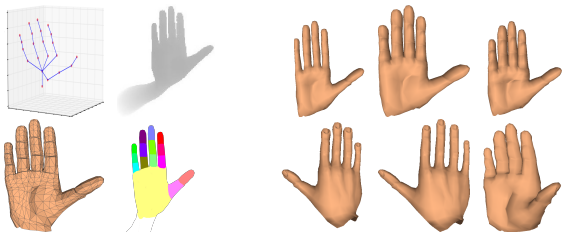
with accurate ground truth joints positions, angles, mesh files, and segmentation masks of depth frames; see Figure 2. Our SynHand5M dataset opens up new possibilities for advanced hand analysis.

Currently, CNN-based discriminative methods are the state-of-the-art which estimate 3D joint positions directly from depth images [8, 21, 4, 27]. However, major weakness of these methods is that the predictions are coarse with no explicit consideration to kinematics and geometric constraints. Sinha et al. [35] propose to estimate 3D shape surface from depth image or hand joint angles, using a CNN. However, their approach neither estimates hand pose nor considers kinematics and physical constraints. Also, these methods generalize poorly to unseen hand shapes [52].

On the other hand, building a personalized hand model requires a different generative approach, that optimizes a complex energy function to generate the hand pose [29, 24, 26, 40, 42]. However, person specific hand model calibration clearly restricts the generalization of these methods for varying hand shapes. Hybrid methods combine the advantages of both discriminative and generative approaches [6, 34, 23, 41, 36, 50, 54]. To the best of our knowledge, none of the existing works explicitly addresses the problem of jointly estimating full hand shape surface, bone-lengths and pose in a single deep framework.

In this paper, we address the problem of generalizing 3D hand pose and surface geometry estimation over varying hand shapes. We propose to embed a novel hand pose and shape layer (HPSL) inside deep learning network to jointly optimize for 3D hand pose and shape surface. The proposed CNN architecture simultaneously estimates the hand pose parameters, bones scales and shape parameters. All these parameters are fed to the HPSL which implements not only a new forward kinematics function, but also the fitting of a morphable hand model and linear blend skinning to produce both 3D joint positions and 3D hand surface; see Figure 3. The whole pipeline is trained in an end-to-end manner. In sum, our contributions are:

1. A novel deep network layer which performs:
   (a) Forward kinematics using a new combination of hand pose and bone scales parameters.
   (b) Reconstruction of a morphable hand model from hand shape parameters and the morph targets.
   (c) Linear blend Skinning algorithm to animate the 3D hand surface; see Section 4.2.
2. A novel end-to-end framework for simultaneous hand pose and shape estimation; see Section 3.
3. A new 5 million scale synthetic hand pose dataset that offers accurate ground truth joint angles, 3D joint positions, 3D mesh vertices, segmentation masks; see Section 5. The synthetic dataset will be publicly available.



(a) Dataset components (b) Shape variations

Figure 2: The SynHand5M dataset contains 5 million images. (a) The dataset ground truth components: hand poses (joints angles and 3D positions), depth maps, mesh files, and hand parts segmentation. (b) Samples illustrating the big variation in shape.

2. Related Work

Depth-based hand pose estimation has been extensively studied in the computer vision community. We refer the reader to the survey [39] for a detailed overview of the field. Recently, a comprehensive analysis and investigation of the state-of-the-art along-with future challenges have been presented by [52]. The approaches can be roughly divided into generative, discriminative and hybrid methods. In this section, we briefly review the existing hand pose benchmarks. Then, we focus our discussion on CNN-based discriminative and hybrid methods.

Existing Benchmarks. Common shortcomings in existing real hand datasets are low variation in hand shape, inaccurate ground truth annotations, insufficient amount of training data, low complexity (e.g. occlusion) of hand poses, and limited view point coverage. Most commonly used benchmarks are NYU [43], ICVL [41] and MSRA15 [38]. NYU hand pose dataset uses a model-based direct search method for annotating ground truth which is quite accurate. It covers a good range of complex hand poses. However, their training set has single hand shape. ICVL dataset uses a guided Latent Tree Model (LTM) based search method and mostly contains highly inaccurate ground truth annotations. Moreover, it uses only one hand model [53]. MSRA15 employs an iterative optimization method [26] for annotating followed by manual refinement. It uses 17 hand poses, however, it has large view-point coverage. The major limitation of this dataset is its limited size and low annotation accuracy. Recently, Yuan et al. [53] propose a million scale real hand pose dataset, but it has low variation in hand shape(i.e. only 10 subjects). Some other very small real hand pose datasets such as Dexter-1 [37], ASTAR [49] , MSRA14 [26] are not suited for large-scale training. Several works focused on creating synthetic hand pose datasets. MSRC [31] is a synthetic benchmark however, it has only one hand model and limited pose space coverage. In [35, 19], medium-scale synthetic hand datasets are used to train CNN models, but they are not publicly

available. Given the hard problem of collecting and annotating a large-scale real hand pose dataset, we propose the first million scale synthetic benchmark which consists of more than 5 million depth images together with ground truth joints positions, angles, mesh files, and segmentation masks.

CNN-based Discriminative Methods. Recent works such as [17, 2, 47, 9, 7, 27] exceed in accuracy over random decision forest (RDF) based discriminative methods [31, 38, 46, 48, 13]. A few manuscripts have used either RGB or RGB-D data to predict 3D joint positions [56, 25, 33, 20]. In [7], Ge et al. directly regress 3D joint coordinates using a 3D-CNN. Recently, [17] introduced voxel-to-voxel regression framework which exploits a one-to-one relationship between voxelised input depth and output 3D heatmaps. [9, 47] introduce a powerful region ensemble strategy which integrates the outputs from multiple regressors on different regions of depth input. Chen et al. [2] extended [47] by an iterative pose guided region ensemble strategy. In [35], a discriminative hand shape estimation is proposed. Although the accuracy of these methods is the state-of-the-art, they impose no explicit geometric and physical constraints on the estimated pose. Also, these methods still fail to generalize on unseen hand shapes [52].

CNN-based Hybrid Methods. Tompson et al. [43] employed CNN for estimating 2D heatmaps. Thereafter, they apply inverse kinematics for hand pose recovery. In extension to this work, [6] utilize 3D-CNN for 2D heatmaps estimation and afterwards regress 3D joint positions. Oberweger et al. [23] utilize three CNNs combined in a feedback loop to regress 3D joint positions. The network comprises of an initial pose estimator, a synthesizer and finally a pose update network. Ye et al. [51] present a hybrid framework using hierarchical spatial attention mechanism and hierarchical PSO. Wan et al. [44] implicitly model the dependencies in the hand skeleton by learning a shared latent space. In [55], a forward kinematics layer, with physical constraints and a fixed hand model, is implemented in an end-to-end training framework. Malik et al. [14] further extend this work by introducing a flexible hand geometry in the training pipeline. The algorithm simultaneously estimates bone-lengths and hand pose. In [45], a multi-task cascade network is employed to predict 2D/3D joint heatmaps along-with 3D joint offsets. Dibra et al. [5] introduce an end-to-end training pipeline to refine the hand pose using an unlabeled dataset. All of the above described methods cast the problem of hand pose estimation to 3D joints regression only. Our argument is that given the inherent 3D surface geometry information in depth inputs, a differentiable hand pose and shape layer can be embedded in the deep learning framework to regress not only the 3D joint positions but also, the full 3D mesh of hand.

3. Method Overview

We aim to jointly estimate the locations of $J = 22$ 3D hand joints , and $\vartheta = 1193$ vertices of hand mesh from a single depth image D_I. Our hand skeleton in rest pose is shown in Figure 3(b). It has J hand joints defined on 26 DoFs. The hand root has 6 DoF; 3 for global orientation and 3 for global translation. All other DoFs are defined for joints articulations. The 26 dimensional pose vector is initialized for the rest pose, called θ_{init}. Any other pose Θ can be constructed by adding change $\delta\theta$ to the rest pose i.e. $\Theta = \theta_{init} + \delta\theta$. The bone-lengths B, are initialized by averaging over all bone-lengths of different hand shapes in our synthetic dataset. In order to add flexibility to the hand skeleton, 6 different hand bones scales, α, are associated to bone-lengths. Our hand mesh has ϑ vertices and 1184 faces. The neutral hand surface is shown in Figure 3(b). We use 7 hand shape parameters β which allow to formulate the surface geometry of a desired hand shape in reference pose; see Section 5.

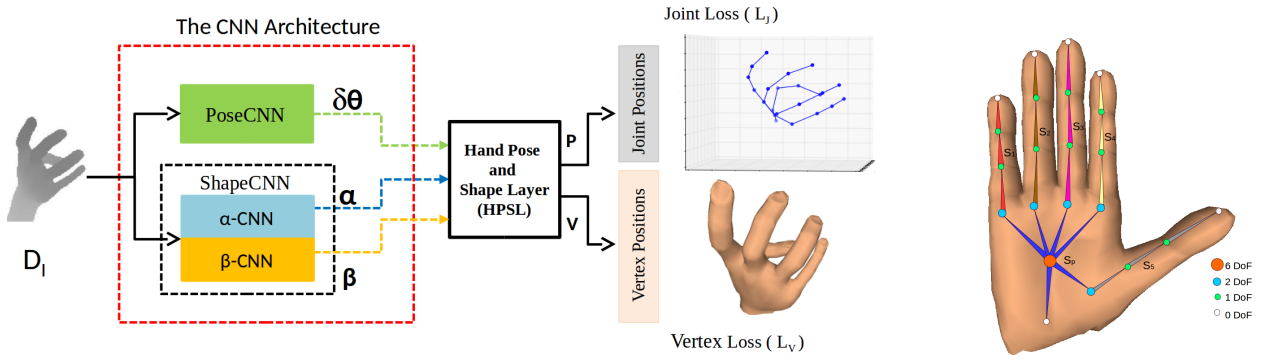
Our pipeline is shown in Figure 3(a). Firstly, a new CNN architecture estimates $\delta\theta$, α and β given a depth input D_I. This architecture consists of PoseCNN which estimates $\delta\theta$ and ShapeCNN which estimates α and β. Thereafter, a new non-linear hand pose and shape layer (HPSL) performs forward kinematics, hand shape surface reconstruction and linear blend skinning. The outputs of the layer are 3D joint positions and hand surface vertices. These outputs are used to compute the standard euclidean loss for joint positions and vertices; see Equation 2. The complete pipeline is trained end-to-end in a fully supervised manner.

4. Joint Hand Pose and Shape Estimation

In this section, we discuss the components of our pipeline which are shown in Figure 3(a). We explain the novel Hand Pose and Shape Layer (HPSL) in detail because it is the main component which allows to jointly estimate hand pose and shape surface.

4.1. The CNN Architecture

Our CNN architecture comprises of three parallel CNNs to learn $\delta\theta$, α and β, given D_I. The PoseCNN leverages one of the state-of-the-art CNN [9] to estimate joint angles $\delta\theta$. However, the CNN was originally used to regress 3D hand joint positions; see Section 2. We refer the reader to [9] for network details of Region Ensemble (REN). In our implementation, the final regressor in REN outputs 26 dimensional $\delta\theta$. The ShapeCNN consists of two simpler CNNs similar to [22]; called α-CNN and β-CNN. Each of them has 3 convolutional layers using kernels sizes 5,5,3 respectively. First two convolution layers are followed by max pool layers. The pooling layers use strides of 4 and 2. The convolutional layers generate 8 feature maps of size 12



(a) Algorithm pipeline (b) Our hand model

Figure 3: (a) An overview of our method for simultaneous 3D hand pose and surface estimation. A depth image D_I is passed through three CNNs to estimate pose parameters $\delta\theta$, bones scales α and shape parameters β. These parameters are sent to HPSL which generate the hand joints positions P and hand surface vertices V. (b) Our hand model with 26 DoFs overlaid with the neutral hand shape b_0. The bone colors illustrate 6 bone-length scales α.

x 12. Lastly, the two fully connected (FC) layers have 1024 neurons each with dropout ratio of 0.3. After the second FC layer, the final FC layers in α-CNN and β-CNN output 6 dimensional α and 7 dimensional β parameters respectively. All layers use the ReLu as activation function.

4.2. Hand Pose and Shape Layer (HPSL)

HPSL is a non-linear differentiable layer, embedded inside the deep network as shown in Figure 3(a). The task of the layer is to produce 3D joint positions $P \in \mathcal{R}^{3xJ}$ and vertices of hand mesh $V \in \mathcal{R}^{3x\vartheta}$ given the pose parameters Θ, hand bones scales α and shape parameters β. The layer function can be written as:

$$(P,V) = \textbf{HPSL}(\Theta, \beta, \alpha) \tag{1}$$

We compute the respective gradients in the layer for back-propagation. The Euclidean 3D joint location and 3D vertex location losses are given as:

$$L_J = \frac{1}{2}\left\|P - P_{GT}\right\|^2 \quad , \quad L_V = \frac{1}{2}\left\|V - V_{GT}\right\|^2 \tag{2}$$

Where L_J and L_V are the 3D joint and vertex losses respectively. P_{GT} and V_{GT} are vectors of 3D ground truth joint positions and mesh vertices, respectively. Various functions inside the layer are detailed as follows:

Hand Skeleton Bone-lengths Adaptation: In order to adapt bone-lengths of hand skeleton during training over varying hand shapes in the dataset, [14] propose various bone-length scaling strategies. Following the similar approach, we assign a separate scale parameter for bone-lengths in palm $\mathbf{s_p}$ and 5 different scales for bones as shown in Figure 3(b). The HPSL acquires the scaling parameters $\alpha = [\mathbf{s_p}, \mathbf{s_1}, \mathbf{s_2}, \mathbf{s_3}, \mathbf{s_4}, \mathbf{s_5}]$ from the ShapeCNN during the training process.

Morphable Hand Model Formulation: Given the shape parameters β learned by our ShapeCNN, we reconstruct the hand shape surface by implementing a morphable hand model inside our HPSL. A morphable hand model $\Psi \in \mathcal{R}^{3x\vartheta}$ is a set of 3D vertices representing a particular hand shape. Any morphable hand model can be expressed as a linear combination of principle hand shape components, called morphable targets $\mathbf{b_t}$ [11]. Our principle hand shape components are defined for *Length*, *Mass*, *Size*, *Palm Length*, *Fingers Inter-distance*, *Fingers Length* and *Fingers Tip-Size*. They represent offsets from a neutral hand shape $\mathbf{b_0}$ similar to one shown in Figure 3(b). Each learned shape parameter $\beta_\mathbf{t}$ defines the amount of contribution of a principle shape components $\mathbf{b_t}$ towards formulation of final hand morphable model. Hence, a hand morphable model Ψ can be formulated using the following Equation:

$$\Psi(\beta) = \mathbf{b_0} + \sum_{\mathbf{t=1}}^{\mathbf{7}} \beta_\mathbf{t}(\mathbf{b_t} - \mathbf{b_0}) \tag{3}$$

Forward Kinematics and Geometric Skinning: To estimate the 3D hand joints positions and surface vertices, we implement forward kinematics and geometric skinning functions inside our HPSL. As this layer is part of our deep network, it is essential to compute and back-propagate the gradients of these functions. The rest of this section addresses the definition of these functions and their gradients.

The deformation of the hand skeleton from the reference pose θ_{init} to the current pose Θ can be obtained by transforming each joint j_i along the kinematic chain by simple rigid transformations matrices. In our algorithm, these matrices are updated based on bones scales α and the changes in pose parameters $\delta\theta$ which are learned by our ShapeCNN and PoseCNN, respectively. The kinematics equation of joint j_i can be written as:

$$\begin{aligned} j_i &= \mathrm{F}_{j_i}(\Theta, \alpha) = \mathrm{M}_{j_i}[0,0,0,1]^T \\ &= \Big(\prod_{k \in S_{j_i}} [\mathrm{R}_{\phi_k}(\theta_k)] \times [\mathrm{T}_{\phi_k}(\alpha B)]\Big)[0,0,0,1]^T \end{aligned} \tag{4}$$

where M_{j_i} represents the transformation matrix from the zero pose (i.e. joint at position $[0,0,0,1]$) to the current pose. S_{j_i} is the set of joints along kinematic chain from j_i to the root joint and ϕ_k is one of the rotation axes of joint k.

For animating the 3D hand mesh, we use linear blend skinning [12] to deform the set of vertices ϑ according to underlying hand skeleton kinematic transformations. The skinning weights ω_i, define the skeleton-to-skin bindings. Their values represent the influence of joints on their associated vertices. Normally, the weights of each vertex are assumed to be convex (i.e. $\sum_{i=1}^{n}\omega_i = 1$) and $\omega_i > 0$. The transformation of a vertex $\mathrm{v}_{\varkappa} \in \Psi$ can be defined as:

$$\begin{aligned}\mathrm{v}_{\varkappa} &= \Upsilon_{\mathrm{v}_{\varkappa}}(\Theta,\beta,\alpha) = \sum_{i\in P_{\mathrm{v}_{\varkappa}}} \omega_i \mathbf{C_{j_i}} \mathrm{v}_{\varkappa}(\beta) \\ &= \sum_{i\in P_{\mathrm{v}_{\varkappa}}} \omega_i \mathbf{C_{j_i}} (\mathbf{b_0^{v_{\varkappa}}} + \sum_{\mathbf{t=1}}^{\mathbf{7}} \beta_{\mathbf{t}} (\mathbf{b_t^{v_{\varkappa}}} - \mathbf{b_0^{v_{\varkappa}}}))\end{aligned} \tag{5}$$

where $P_{\mathrm{v}_{\varkappa}}$ is the set of joints influencing the vertex $\mathrm{v}_{\varkappa}$ and $\mathbf{C_{j_i}}$ is the transformation matrix of each joint j_i from its reference pose θ_{init} to its actual position in the current animated posture. $\mathbf{C_{j_i}}$ can be represented as:

$$\mathbf{C_{j_i}} = \mathrm{M_{j_i}} \mathrm{M_{j_i}^{*-1}} \tag{6}$$

where $\mathrm{M}_{j_i}^{*-1}$ defines the inverse of reference pose transformation matrix.

Gradients computation: For backward-pass in the **HPSL**, we compute gradients of the following equation with respect to the layer inputs:

$$\mathbf{HPSL}(\Theta,\beta,\alpha) = (\, \boldsymbol{F}(\Theta,\alpha)\,, \Upsilon(\Theta,\beta,\alpha)\,). \tag{7}$$

Each vertex $\mathrm{v}_{\varkappa} = \mathrm{HPSL}_{\mathrm{v}_{\varkappa}}(\Theta,\beta,\alpha)$ in the reconstructed hand morphable model Ψ is deformed using Equation 5. Hence, its gradients with respect to a shape parameter β_t can be computed as:

$$\frac{\partial(\mathrm{HPSL}_{\mathrm{v}_{\varkappa}})}{\partial\beta_t} = \sum_i \omega_i \mathbf{C_{j_i}} (\mathbf{b_t^{v_{\varkappa}}} - \mathbf{b_0^{v_{\varkappa}}}) \quad \text{for } t = 1,2,\ldots,7$$

According to Equation 7, bones scales influence the joints positions and vertices positions. Hence, the resultant gradient with respect to a hand scale parameter α_s, can be calculated as:

$$\frac{\partial(\mathbf{HPSL})}{\partial\alpha_s} = \frac{\partial\mathbf{F}}{\partial\alpha_s} + \frac{\partial\Upsilon}{\partial\alpha_s} \qquad \text{for } s = 1,2,\ldots,6$$

To compute the partial derivative of $\mathbf{F}$ with respect to α_s, we need to derivate each joint with respect to its associated scale parameter. The gradient of a joint with respect to α_s, can be computed by replacing the scaled translational matrix containing α_s by its derivative and keep all other matrices same; see Equation 2 in supplementary document. In a similar way, the gradient of a vertex $\mathrm{v}_{\varkappa}$ with respect to α_s can be computed by:

$$\begin{aligned}\frac{\partial\Upsilon_{\mathrm{v}_{\varkappa}}}{\partial\alpha_s} &= \sum_i \omega_i \frac{\partial\mathbf{C_{j_i}}}{\partial\alpha_s} \mathrm{v}_{\varkappa} \\ &= \sum_i \omega_i [\mathrm{M}_{j_i} (\mathrm{M}_{j_i}^{*-1})' + (\mathrm{M}_{j_i})' \mathrm{M}_{j_i}^{*-1}] \mathrm{v}_{\varkappa}\end{aligned}$$

Likewise, for the pose parameters Θ, we compute the following equation:

$$\frac{\partial(\mathbf{HPSL})}{\partial\theta_p} = \frac{\partial\mathbf{F}}{\partial\theta_p} + \frac{\partial\Upsilon}{\partial\theta_p} \qquad \text{for } p = 1,2,\ldots,26$$

Accordingly, the derivative of a joint with respect to a pose parameter θ_p, is simply to replace the rotation matrix of θ_p by its derivation; see Equation 5 in supplementary document. And, the derivative of a vertex $\mathrm{v}_{\varkappa}$ with respect to θ_p is computed by:

$$\begin{aligned}\frac{\partial\Upsilon_{\mathrm{v}_{\varkappa}}}{\partial\theta_p} &= \sum_i \omega_i \frac{\partial\mathbf{C_{j_i}}}{\partial\theta_p} \mathrm{v}_{\varkappa} \\ &= \sum_i \omega_i [(\mathrm{M}_{j_i})' \mathrm{M}_{j_i}^{*-1}] \mathrm{v}_{\varkappa} \quad \text{for } p = 1,2,\ldots,26\end{aligned}$$

More details about the gradients computation can be found in the supplementary document.

5. Synthetic Dataset

There are two main objectives of creating our synthetic dataset. First is to jointly recover full hand shape surface and pose provided that there is no ground truth hand surface information available in public benchmarks; see Section 6.2. Second objective is to provide a training data with sufficient variation in hand shapes and poses such that a CNN model can be pre-trained to improve the recognition rates on real benchmarks; see Section 6.3. This problem is different from generating very realistic hand-shape, where a real-world statistical hand model [30] can be applied. However, the variation in shape is more challenging for real-world databases e.g. BigHand2.2M [53] database was captured from only 10 users, and the MANO [30] database was built from the contribution of 31 users. Instead, we generate a bigger hand shape variation which may not be present in a given cohort of human users.

Our SynHand5M dataset offers $4.5M$ train and $500K$ test images; see Figure 2(a) for SynHand5M components. SynHand5M uses the hand model generated by MakeHuman-BastionLAB [15] which is a procedural full-body generator distributed as add-on of the Blender [1] 3D authoring software. Our virtual camera simulates a Creative Senz3D Interactive Gesture Camera [3]. It renders images of resolution 320x240 using diagonal field of view of 74 degrees.

In the default position, the hand palm faces the camera orthogonally and the fingers point up. We procedurally modulate many parameters controlling the hand and generate images by rendering the view from the virtual camera. The parameters characterizing the hand model belong to three categories: hand shape, pose and view point.

Without constraints the hand generator can easily lead to impossible hand shapes. So, in order to define realistic range limits for modulating hand shapes, we relied on the DINED [16] anthropometric database. DINED is a repository collecting the results of several anthropometric databases, including the CAESAR surface anthropometry survey [28]. We manually tuned the ranges of the 7 hand shape parameters (see Section 4.2) in order to cover 99% of the measured population in this dataset; see supplementary document for more details.

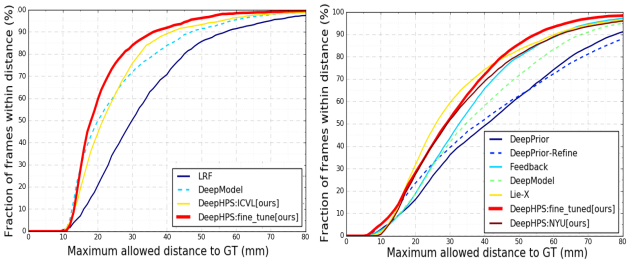
To modulate the hand pose, we manipulate the 26 DoFs of our hand model; see Figure 3(b). For each finger, rotations are applied to flexion of all phalanges plus the abduction of the proximal phalanx. Additionally, in order to increase the realism of the closed fist configuration, the roll of middle, ring, and pinky fingers is derived from the abduction angle of the same phalanx. The rotation limits are set to bring the hand from a closed fist to an over-extended aperture, respecting anatomical constraints and avoiding the fingers to enter the palm.

The hand can rotate about three DoFs to generate different view points: roll around its longitudinal axis (i.e. along the fingers), rotate around the palm orthogonal axis (i.e. rolling in front of the camera), and rotate around its transversal axis (i.e. flexion/extension of the wrist).

6. Experiments and Results

In this section, we provide the implementation details, quantitative and qualitative evaluations of the proposed algorithm and the proposed dataset. We use three evaluation metrics; mean 3D joint location error (JLE), 3D vertex location error (VLE) and percentage of images within certain thresholds in mm.

Recent CNN-based discriminative methods such as [7, 47, 17, 27] outperform CNN-based hybrid methods; see Section 2. However, due to direct joints regression, discriminative methods neither explicitly account for the hand shapes nor consider kinematics constraints [55, 14]. Moreover, in contrast to hybrid methods, discriminative methods generalize poorly to unseen hand shapes; see [52]. Our proposed hybrid method does not exceed in accuracy over recent discriminative works but, it does not suffer from such limitations. Therefore, it is not fair to compare with these methods. However, we compare with the state-of-the-art hybrid methods and show improved performance. Notably, we propose the first algorithm that jointly regresses hand pose, bone-lengths and shape surface in a single network.



(a) ICVL (b) NYU

Figure 4: **Quantitative evaluation**. (a) show the results of our algorithm (DeepHPS) on ICVL test set, when trained on ICVL and fine-tuned on ICVL. (b) is the same but with NYU. To fine-tune, we pretrain DeepHPS on our SynHand5M. Our results on ICVL show improved accuracy over the state-of-the-art hybrid methods (e.g. LRF[41] and DeepModel[55]). On NYU, the results are better than the state-of-the-art hybrid methods (e.g. DeepPrior[22], DeepPrior-Refine[22], Feedback[23], DeepModel[55] and Lie-X[48]). The curves show the number of frames in error within certain thresholds.

6.1. Implementation Details

For training, we pre-process the raw depth data for standardization and depth invariance. We start by computing the centroid of the hand region in the depth image. The obtained 3D hand center location (i.e. palm center) is used to crop the depth frame. The camera intrinsics (i.e. focal length) and a bounding box of size 150, are used during the crop. The pre-processed depth image is of size 96 x 96 and in depth range of $[-1, 1]$. The annotations in camera coordinates are simply normalized by the bounding box size and clipped in range $[-1, 1]$.

We use Caffe [10] which is an open-source training framework for deep networks. The complete pipeline is trained end-to-end until convergence. The learning rate was set to 0.00001 with 0.9 SGD momentum. A batch size of 256 was used during the training. The framework is executed on a desktop equipped with Nvidia Geforce GTX 1080 Ti GPU with 16GB RAM. One forward pass takes $3.7ms$ to generate 3D hand joint positions and shape surface. For simplicity, we name our method as DeepHPS.

6.2. Algorithm Evaluation

In this subsection, we evaluate our complete pipeline using the SynHand5M. Moreover, we devise a joint training strategy for both real and synthetic datasets to show qualitative hand surface reconstruction of real images.

Evaluation on the synthetic dataset: The complete pipeline is trained end-to-end using SynHand5M for pose and shape recovery. For fair comparison, we train the state-of-the-art model based learning methods [55, 14] on SynHand5M. [14] works for varying hand shapes in contrast to

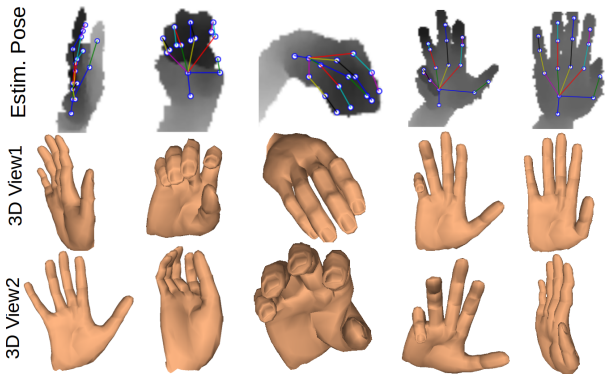


Figure 5: **Real hand pose and shape recovery**: More results on hand pose and surface reconstruction of NYU[43] images. Despite of unavailability of ground truth hand mesh vertices, our algorithm produces plausible hand shape.

the closely related method [55]. The quantitative results are shown in Table 1. Our method clearly exceeds in accuracy over the compared method and additionally reconstructs full hand surface. The qualitative results are shown in Figure 6. The estimated 22 joint positions are overlaid on the depth images while the reconstructed hand surface is shown using two different views named as ***3D View1*** and ***3D View2***. For better visualization, view2 is similar to ground truth view. The results demonstrate that our DeepHPS model infers correct hand shape surface even in cases of occlusion of several fingers and large variation in view points.

Evaluation on the NYU real dataset: In order to jointly train our whole pipeline on both real and synthetic data, we found 16 closely matching common joint positions in SynHand5M and the NYU dataset. These common joints are different from the 14 joints used for the public comparisons [43]. The loss equation is;

$$L = L_J + \mathbb{1}L_V \tag{8}$$

where $\mathbb{1}$ is an indicator function which specifies whether the ground truth for mesh vertices is available or not. In our setup, it is 1 for synthetic images and 0 for real images. For real images, backpropagation from surface reconstruction part is disabled.

The qualitative pose and surface shape results on sample NYU real images are shown in Figure 1 and 5. Despite of the missing ground truth surface information and presence of high camera noise in NYU images, the resulting hand surface is plausible and the algorithm performs well in case of missing depth information and occluded hand parts.

6.3. Comparison on Public Benchmarks

The public benchmarks do not provide ground truth hand mesh files. Therefore, we provide quantitative results for pose inference on two of the real hand pose datasets (i.e. NYU and ICVL). For comparisons, NYU dataset use 14 joint positions [43] whereas ICVL dataset [41] use 16 joint positions.

Method \Error(mm)	3D Joint Loc.	3D Vertex Loc.
DeepModel [55]	11.36	–
HandScales [14]	9.67	–
DeepHPS [**Ours**]	**6.3**	**11.8**

Table 1: **Quantitative Evaluation on SynHand5M**: We show the 3D joint and vertex locations errors(mm). Our method additionally outputs mesh vertices and outperforms model based learning methods [55, 14].

Methods	3D Joint Location Error
DeepPrior [22]	20.75mm
DeepPrior-Refine [22]	19.72mm
Crossing Nets [44]	15.5mm
Feedback [23]	15.9mm
DeepModel [55]	17.0mm
Lie-X [48]	14.5mm
DeepHPS:NYU [Ours]	15.8mm
DeepHPS:fine-tuned [**Ours**]	**14.2mm**

Table 2: **Quantitative comparison on NYU** [43]: Our fine-tuned DeepHPS model on the NYU dataset shows the state-of-the-art performance among hybrid methods.

Methods	3D Joint Location Error
LRF [41]	12.57mm
DeepModel [55]	11.56mm
Crossing Nets [44]	10.2mm
DeepHPS:ICVL [Ours]	10.5mm
DeepHPS:fine-tuned [**Ours**]	**9.1mm**

Table 3: **Quantitative comparison on ICVL** [41]: The DeepHPS model fine-tuned on the ICVL dataset outperforms the state-of-the-art hybrid methods.

Our DeepHPS algorithm is trained on NYU and ICVL individually, called DeepHPS:NYU and DeepHPS:ICVL models. Then, we fine-tune the pre-trained DeepHPS (on SynHand5M) with the NYU and ICVL, we call DeepHPS:fine-tuned models. The 3D joint location errors of the trained models are calculated on 8252 NYU and 1596 ICVL test images respectively. The quantitative results are shown in Figure 4 and Tables 2 and 3. DeepHPS:fine-tuned models achieve an error improvement of 13.3% and 10.12% over DeepHPS:ICVL and DeepHPS:NYU models respectively.

On the ICVL and NYU datasets, we achieve improvement in the joint location accuracy over the state-of-the-art hybrid methods.

Failure case: Our framework works well in case of missing depth information and occlusions. However, under severe occlusions and a lot of missing depth information, it may fail to detect the correct pose and shape; see Figure 7.

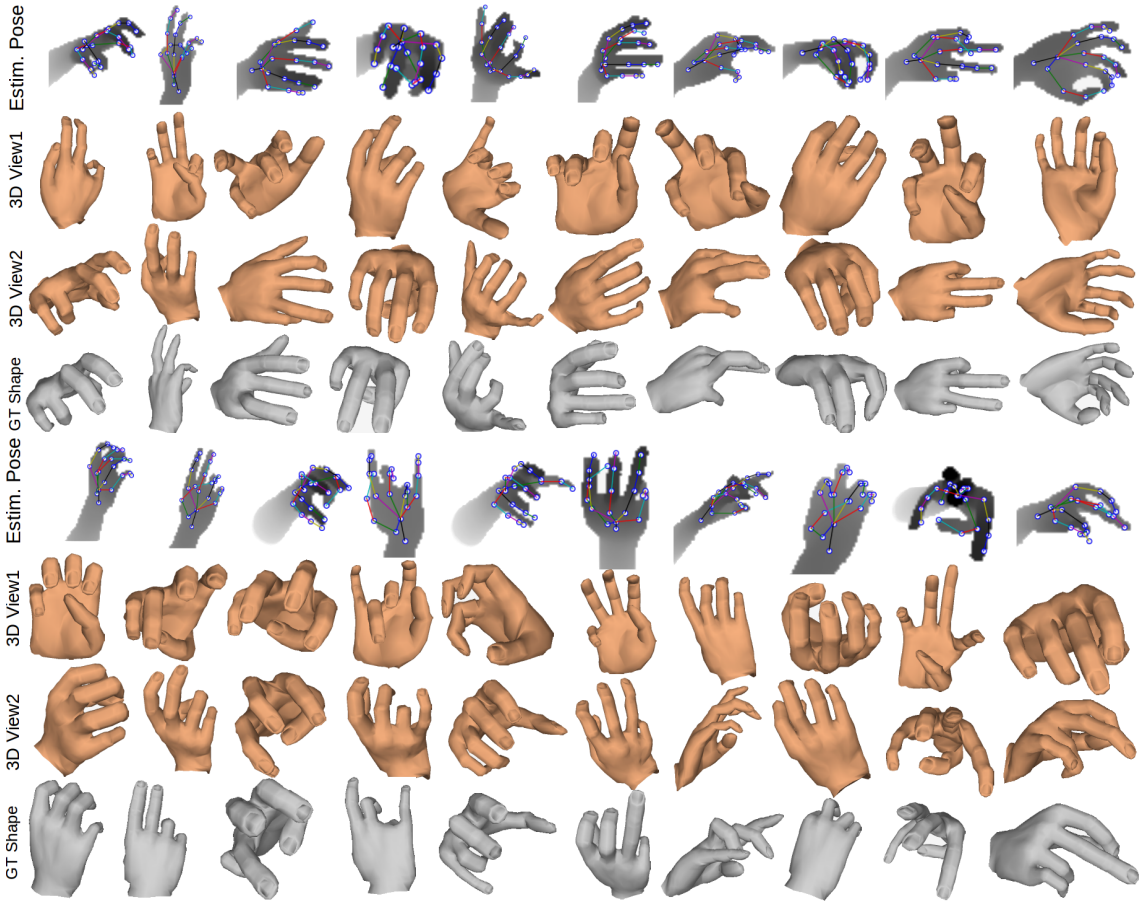
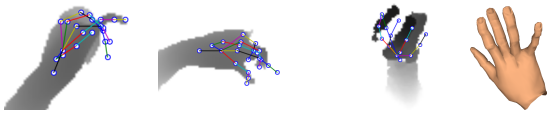


Figure 6: **Synthetic hand pose and shape recovery**: We show example estimated hand poses overlaid with the preprocessed depth images from our SynHand5M. We show the reconstructed surface from two different views (yellow) and the ground truth surface (gray). ***3D View2*** is similar to the ground truth view. Our algorithm infers correct 3D pose and shape even in very challenging condition, like occlusion of several fingers and large variation in view points.



(a) (b)

Figure 7: **Failure case**:(a) incorrect pose due to highly occluded hand parts. (b) incorrect pose and shape due to significant missing depth information.

7. Conclusion and Future Work

In this work, we demonstrate the simultaneous recovery of hand pose and shape surface from a single depth image. For training, we synthetically generate a large scale dataset with accurate joint positions, segmentation masks and hand meshes of depth images. Our dataset will be a valuable addition for training and testing CNN-based models for 3D hand pose and shape analysis. Furthermore, it improves the recognition rate of CNN models on hand pose datasets. In our algorithm, intermediate parametric representations are estimated from a CNN architecture. Then, a novel hand pose and shape layer is embedded inside the deep network to produce 3D hand joint positions and shape surface. Experiments show improved accuracy over the state-of-the-art hybrid methods. Furthermore, we demonstrate plausible results for the recovery of hand shape surface on real images. Improving the performance of CNN-based hybrid methods is a potential research direction. These methods bear a lot of potential due to their inherent stability and scalability. In future, we wish to extend our dataset with wider view points coverage, object interactions and RGB images. Another aspect for future work is predicting fine-scale 3D surface detail on the hand, where real-world statistical hand models [30] possibly give better priors.

Acknowledgements

This work was partially funded by NUST, Pakistan, the Federal Ministry of Education and Research of the Federal Republic of Germany as part of the research projects DYNAMICS (Grant number 01IW15003) and VIDETE (Grant number 01IW18002).

References

[1] Blender. https://www.blender.org, March 2018. 5

[2] X. Chen, G. Wang, H. Guo, and C. Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *arXiv preprint arXiv:1708.03416*, 2017. 3

[3] Creative. Senz3d interactive gesture camera. https://us.creative.com/p/web-cameras/creative-senz3d, March 2018. 5

[4] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang. Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224*, 2017. 2

[5] E. Dibra, T. Wolf, C. Oztireli, and M. Gross. How to refine 3d hand pose estimation from unlabelled depth data? *In 3DV*, 2017. 3

[6] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3593–3601, 2016. 2, 3

[7] L. Ge, H. Liang, J. Yuan, and D. Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 6

[8] H. Guo, G. Wang, and X. Chen. Two-stream convolutional neural network for accurate rgb-d fingertip detection using depth and edge information. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2608–2612. IEEE, 2016. 2

[9] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang. Region ensemble network: Improving convolutional network for hand pose estimation. *In ICIP*, 2017. 3

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 6

[11] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. H. Pighin, and Z. Deng. Practice and theory of blendshape facial models. 4

[12] 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*, pages 165–172. ACM Press/Addison-Wesley Publishing Co., 2000. 5

[13] P. Li, H. Ling, X. Li, and C. Liao. 3d hand pose estimation using randomized decision forest with segmentation index points. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 819–827, 2015. 3

[14] J. Malik, A. Elhayek, and D. Stricker. Simultaneous hand pose and skeleton bone-lengths estimation from a single depth image. *In 3DV*, 2017. 3, 4, 6, 7

[15] ManuelBastioni. v1.5.0. http://www.manuelbastioni.com, March 2018. 5

[16] J. Molenbroek. Dined, anthropometric database. https://dined.io.tudelft.nl/, 2004. 6

[17] 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. *arXiv preprint arXiv:1711.07399*, 2017. 3, 6

[18] 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. *arXiv preprint arXiv:1712.01057*, 2017. 1

[19] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1163–1172, 2017. 2

[20] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 10, 2017. 3

[21] M. Oberweger and V. Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV workshop*, volume 840, page 2, 2017. 2

[22] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *In CVWW*, 2015. 3, 6, 7

[23] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324, 2015. 2, 3, 6, 7

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

[25] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. *arXiv preprint arXiv:1712.03866*, 2017. 3

[26] 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

[27] M. Rad, M. Oberweger, and V. Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. *arXiv preprint arXiv:1712.03904*, 2017. 2, 3, 6

[28] K. Robinette, H. Daanen, and E. Paquet. The CAESAR project: a 3-D surface anthropometry survey. pages 380–386. IEEE Comput. Soc, 1999. 6

[29] K. Roditakis, A. Makris, and A. Antonis. Generative 3d hand tracking with spatially constrained pose sampling. In *In BMVC*. IEEE, 2017. 2

[30] 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):245:1–245:17, Nov. 2017. 1, 5, 8

[31] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015. 2, 3

[32] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6, 2017. 1

[33] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 3

[34] A. Sinha, C. Choi, and K. Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4150–4158, 2016. 2

[35] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proc. CVPR*, 2017. 2, 3

[36] 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 *European Conference on Computer Vision*, pages 294–310. Springer, 2016. 2

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

[38] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015. 2, 3

[39] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *IEEE international conference on computer vision*, pages 1868–1876, 2015. 2

[40] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, volume 34, pages 101–114. Wiley Online Library, 2015. 2

[41] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014. 2, 6, 7

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

[43] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014. 1, 2, 3, 7

[44] C. Wan, T. Probst, L. Van Gool, and A. Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 3, 7

[45] C. Wan, T. Probst, L. Van Gool, and A. Yao. Dense 3d regression for hand pose estimation. *arXiv preprint arXiv:1711.08996*, 2017. 3

[46] C. Wan, A. Yao, and L. Van Gool. Hand pose estimation from local surface normals. In *European Conference on Computer Vision*, pages 554–569. Springer, 2016. 3

[47] G. Wang, X. Chen, H. Guo, and C. Zhang. Region ensemble network: Towards good practices for deep 3d hand pose estimation. *Journal of Visual Communication and Image Representation*, 2018. 3, 6

[48] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng. Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *International Journal of Computer Vision*, pages 1–25, 2017. 3, 6, 7

[49] C. Xu, A. Nanjappa, X. Zhang, and L. Cheng. Estimate hand poses efficiently from single depth images. *International Journal of Computer Vision*, 116(1):21–45, 2016. 2

[50] Q. Ye and T.-K. Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. *arXiv preprint arXiv:1711.10872*, 2017. 2

[51] Q. Ye, S. Yuan, and T.-K. Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *European Conference on Computer Vision*, pages 346–361. Springer, 2016. 3

[52] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *IEEE CVPR*, 2018. 2, 3, 6

[53] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2605–2613. IEEE, 2017. 1, 2, 5

[54] Y. Zhang, C. Xu, and L. Cheng. Learning to search on manifolds for 3d pose estimation of articulated objects. *arXiv preprint arXiv:1612.00596*, 2016. 2

[55] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. *In IJCAI*, 2016. 3, 6, 7

[56] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *International Conference on Computer Vision*, 2017. 3