Weakly-supervised Learning for Single Depth based Hand Shape Recovery

Xiaoming Deng, Yuying Zhu, Yinda Zhang, Zhaopeng Cui, Ping Tan,
Wentian Qu, Cuixia Ma, and Hongan Wang

***Abstract*—Recent emerging technologies such AR/VR and HCI are drawing high demand on more comprehensive hand shape understanding, requiring not only 3D hand skeleton pose but also hand shape geometry. In this paper, we propose a deep learning framework to produce 3D hand shape from a single depth image. To address the challenge that capturing ground truth 3D hand shape in the training dataset is non-trivial, we leverage synthetic data to construct a statistical hand shape model and adopt weak supervision from widely accessible hand skeleton pose annotation. To bridge the gap due to the different hand skeleton definitions in the existing public datasets, we propose a joint regression network for hand pose adaptation. To reconstruct the hand shape, we use Chamfer loss between the predicted hand shape and the point cloud from the input depth to learn the shape reconstruction model in a weakly-supervised manner. Experiments demonstrate that our model adapts well to the real data and produces accurate hand shapes that outperform the state-of-the-art methods both qualitatively and quantitatively.**

Hand pose estimation that aims to recover hand skeleton joint positions is important for many applications, such as AR/VR and HCI, and it has been studied for decades. However, compared to hand pose estimation, the problem of reconstructing full 3D hand shape model that consists of 3D vertices, edges and faces between vertices has been rarely studied [1], [2]. Compared to unordered point clouds, hand shape model generally has the same geometric topology of vertices for different subjects. This task is on high demand by many applications involving accurate interaction in virtual environment, which requires not only skeleton pose like many previous work [3], [4] but also 3D shape that models the hand skin. Hand shape recovery is essential to enhance experience in AR/VR applications, especially, the modeling of hand-object surface contacts when interacting with objects [5].

Manuscript received December 27, 2019; revised May 27, 2020, August 22, 2020, and September 23, 2020; accepted October 14, 2020. Date of publication XX XX, 2020; date of current version XX XX, 2020. This work was supported in part by the National Key R&D Program of China under Grant 2016YFB1001201, in part by the National Natural Science Foundation of China under Grant 61473276 and Grant 61872346, in part by the Natural Science Foundation of Beijing under Grant L182052, and in part by the Distinguished Young Researcher Program, Institute of Software, Chinese Academy of Sciences. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Guo-Jun Qi. *Corresponding authors: X. Deng; Y. Zhang; Z. Cui*

X. Deng, Y. Zhu, W. Qu, C. Ma, H. Wang are with the Beijing Key Laboratory of Human Computer Interactions, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: {xiaoming, cuixia, hongan}@iscas.ac.cn)

Y. Zhang is with Google, USA. (e-mail: yindaz@google.com)

Z. Cui is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China. (e-mail: zhpcui@gmail.com)

P. Tan is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: pingtan@sfu.ca)

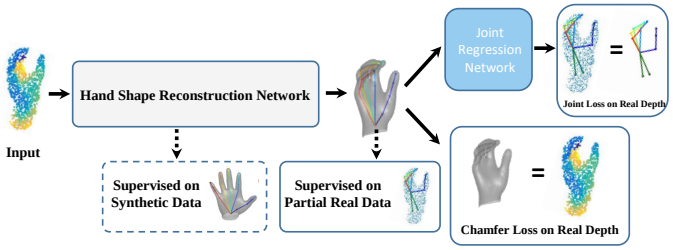


Fig. 1. Overview of our network for hand shape recovery. Our method takes point cloud from depth as input and generates the 3D hand shape model. The hand shape reconstruction network recovers hand shape and joints with a statistical hand model. The predicted hand shape is fed into joint regression network to produce hand joints on the real data.

Besides challenges inherited from hand pose estimation, such as occlusions and high degree of freedom, the lack of ground truth on real data acts as one of the major difficulty for 3D hand shape generation. One plausible solution is to use synthetic data. However, it is not ideal due to the notorious domain gap issue [6], [7]. In one of the most recent effort, Malik *et al.* [1] proposed a deep learning model, named DeepHPS, to learn hand shape and hand skeleton pose at the same time. While hand pose and shape can be learned under full supervision from the synthetic data, and the 3D hand pose is further trained on real data with annotated ground truth hand pose, the quality of the reconstructed hand shape is restricted by the lack of ground truth hand shape and the gap between the skeleton of predefined model and those of real datasets.

Another challenge is due to the domain gap issue. On one hand, the hand pose skeleton definition varies from one dataset to another, which prevents the pre-trained model on a synthetic data from being finetuned on real datasets. For example, the NYU dataset [3] uses a hand skeleton consisting 36 joints; however the pose of ICVL dataset ICVL [8] has only 16 joints. Malik *et al.* [1] proposed to find an intersection set of hand joints between the synthetic and real data, however this intersection set only maintains partial supervision or may not even exist in some case. How to effectively transfer 3D hand pose knowledge from synthetic to real data under different hand pose definitions is still an open problem. On the other hand, it is even harder to transfer hand shape knowledge across the domain. Malik *et al.* [1] does not handle this part on real data and bets on the generalization capability of the network, and as a result the shape on real data often comes with mean shape and lack of details.

In this paper, we propose a new deep learning framework

to predict hand shape model from a single depth image (Fig. 1). To fully exploit pose annotations on real data with a skeleton pose definition that might be different from that of the synthetic data for pretraining, we design an auxiliary joint regression network that takes our reconstructed 3D hand shape as input to produce 3D hand skeleton joint on the real data. This auxiliary network allows us to define supervised loss on all the hand joints in the real data pose definition, which effectively back-propagates gradient to the generated hand shape. To learn shape on real image, we use Chamfer loss between reconstructed hand shape and point cloud from the input depth as the supervision. This encourages the model to learn more detailed shape parameters, and produces the 3D hand shape that is consistent with the input depth. Our method is a weakly-supervised model due to the fact that we do not adopt supervision on the hand shape and only use hand joints and input point clouds as supervision.

The main contributions of this work are as follows:

1) We propose a deep learning framework for hand shape reconstruction with easy-to-access weak supervisions of hand pose and point cloud from the input depth;
2) We present a joint regression network, which uses hand shape as input to predict hand joints and facilitates the hand pose adaptation of different hand skeleton definitions;
3) Extensive experiments demonstrate that our method can produce accurate hand shape, and outperforms the state-of-the-art hand shape recovery methods in both hand pose and shape accuracy.

I. Related Work

Hand Shape Representation The hand shape can be represented with a combination of pose and shape parameters. The pose parameters encode the location of hand joints, and the shape parameters measure geometric details such as the perimeter of the finger and thickness of the palm. Surface mesh [9], [10] builds a linear shape basis by parameterizing the variation of hand mesh in the neural pose, and adopts joint rotations to describe the variation of hand pose. MANO [11] borrows the idea from the famous human body model SMPL [12] for hand, which is more accurate by introducing pose dependent shape variation. Tagliasacchi *et al.* [13] introduce a compact representation using sphere mesh representation, which eases the collision detection task. In this paper, we adopt a statistical hand model using SMPL model in our hand shape reconstruction network, but the other similar hand models such as MANO could be also used. We use SMPL for our hand shape model, since SMPL could balance the representation accuracy and efficiency well.

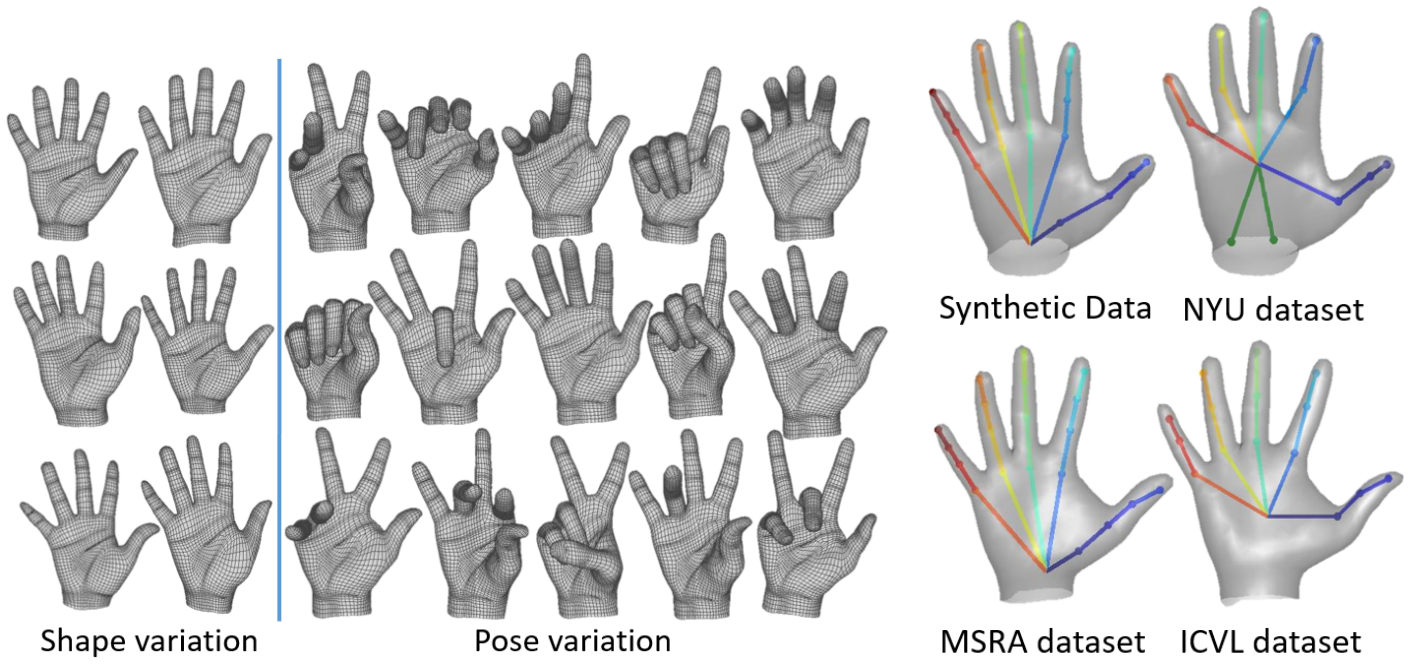
3D Hand Pose Estimation CNN based hand pose estimation methods [14], [15], [6], [16], [17], [18] have progressed rapidly and achieved excellent performance. Most of CNN based methods [18], [19] get 3D hand joint locations by regression, and a few efforts [3], [4], [20] estimate 3D hand pose via 2D or 3D heatmaps for each joints using per-pixel or per-voxel classification. Recently, CNN based methods using 3D representations of depth, such as voxel [21], [4], [17], point cloud [22], [23] and graph [24], [25], adopt 3D CNN, PointNet++ or Graph Convolutional Networks to regress hand pose. We use point cloud to represent depth, and adopt PointNet++ as backbone to recover hand shape and pose, since PointNet++ is effective to achieve accurate 3D hand pose related tasks [22], [23]. If the 3D point cloud of a depth image is represented with graph, unsupervised representation learning methods such as GraphTER [25] may be used to reduce the high demand of label data.

Hand Shape Recovery Taylor *et al.* [26] recover personalized hand models with a depth sequences where the user's hand rotates 180 degrees whilst articulating fingers. Khamis *et al.* [9] propose a hand model, and the parameters are trained jointly on large-scale dataset. The key idea is to refine a cost function that encourages the observed depth map to be explained by the proposed articulated model. Following the spirit of [9], Tan *et al.* [10] introduce a render-and-compare loss and use Levenberg-Marquardt with finite difference approximation to accelerate optimization. The above methods require several frames to initialize hand model. In hand tracking problem, Taylor *et al.* [27] and Tagliasacchi *et al.* [13] both use the results of the previous frame as an initialization for current frame. Though model-based methods can achieve pretty good hand shapes, they require high-quality initialization to avoid local minimal. Li *et al.* [28] conduct a self-supervised method to learn linear blend skinning (LBS) and model fitting of articulated shapes to point clouds. Malik *et al.* [1] propose the first CNN network to recovery hand shape from single depth. However, the quality of the reconstructed hand shape using [1] is restricted by the gap between the skeleton of predefined model and those of real datasets. Recently, Ge *et al.* [29] propose a method to recover 3d hand shape and pose estimation from a single color image. The method requires pretraining on synthetic data and leads to incorrect hand shape without pretraining. Different to previous methods, our method can learn hand shape by transferring knowledge from synthetic data to real data with supervision of hand pose, and use an auxiliary network and Chamfer loss to bridge the gaps in both hand shape and pose.

II. Synthetic Hand Shape Dataset

In order to build statistical hand shape model, a large scale of hand shape including sufficient variation in hand shape and pose is necessary. However, it is a very tedious task to capture hand shape even with the most advanced hand-held 3D scanners such as Artec Spider scanner [30]. Moreover, public hand pose datasets only contain hand joints, and it is time-consuming to label 3D hand shape manually. Alternatively, synthetic dataset is a practical way to solve the data deficiency problem. The synthetic data plays a crucial role for learning hand shape reconstruction model, and the diversity and realism are important for the generalization capability of the model.

We first create a large high-fidelity synthetic hand shape dataset. We use an Artec Eva scanner to collect hand shapes under standard pose from 50 subjects, which are significantly more realistic than CAD models as used in other datasets [31], [1]. We then use a nonrigid ICP [32] to register all



(a) Our hand shape dataset (b) Skeleton models

Fig. 2. (a) Our synthetic dataset with hand shape and pose variations. (b) The hand skeleton joint models in our synthetic dataset, NYU, MSRA and ICVL datasets.

scanned hand shapes such that they have exactly the same number of vertices and topology. This facilitates us to deform hand shapes from the neutral pose to arbitrary pose, and maintain the correspondence. To obtain a collection of realistic poses, we summarize 1840 key hand poses from popular datasets such NYU dataset [3] and BigHand2.2M dataset [33], which include most hand interaction gestures. These poses are applied to a standard skeleton of each subject, and we employ professional artists to bind these poses with the hand shape model using Maya [34]. Hand shapes of each subject are deformed according to the poses of its corresponding skeleton.

Fig. 2 (a) shows several examples of our synthetic dataset. We get a variety of hand shape models under different shape and pose variations. Then we use Mitsuba [35] to render depth images under different camera views. In total, we get 270K pairs of depth map with ground truth hand shape and joint positions. In our dataset, we have $S = 50$ subjects, each subject contains $P = 1840$ poses, and each hand data has $N = 1305$ vertices and $K = 21$ joints.

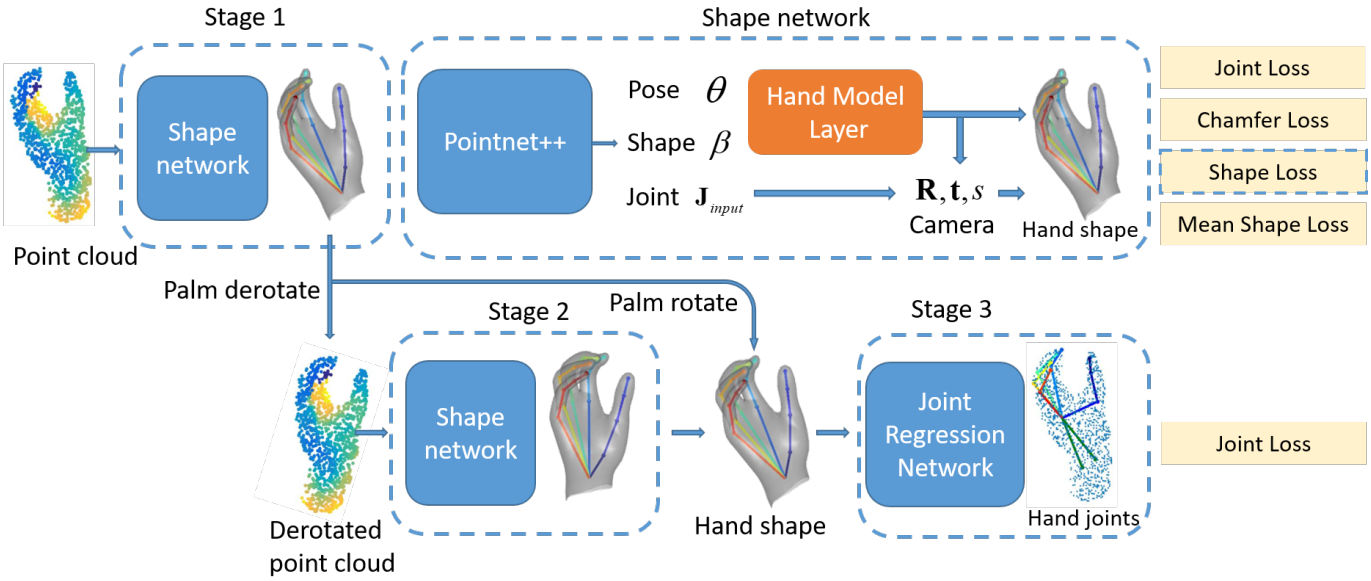
III. Approach

As illustrated in Fig. 3(a), we propose a neural network to reconstruct hand shape model from the point cloud from a single depth image. Our network first adopts a hand shape reconstruction network (Stage 1 and Stage 2 in Fig. 3(a)) to predict hand shape and its binding joints from the hand point cloud via input depth. Then we design a joint regression network to produce 3D hand joints on the real data with our generated hand shape as input. Joint regression network bridges the gap of hand pose in the synthetic and real datasets, and it also back-propagates gradient to the generated shape, providing a weak supervision for hand shape. Chamfer loss between the reconstructed hand shape and the point cloud from an input depth image provides an extra weak supervision for hand shape, which encourages to produce 3D hand shape that is consistent with the input point cloud.

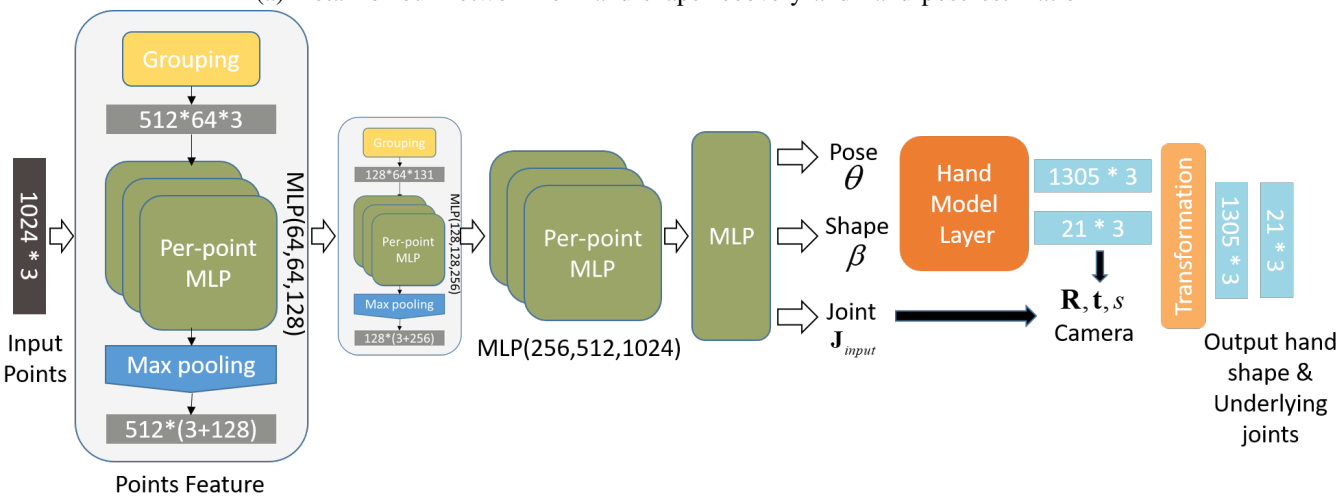
A. Hand Shape Reconstruction Network

We adopt a two-stage hand shape network to generate hand shape model (shown as Stage 1 and Stage 2 in Fig. 3(a)). The hand shape network uses the point cloud as input, and it is built upon PointNet++ [36], which has been proved as an effective 3D representation for hand pose estimation [22], [23], [16]. The point cloud is fed to PointNet++ with three point abstraction levels, the last level extracts 1024-d global features, then we use three parallel branch networks to regress hand shape parameter β, hand pose parameter θ, and compute the 3D transformation between the camera and hand coordinate system consisting rotation matrix $\mathbf{R}$, translation vector $\mathbf{t}$ and global scale s. The global scale s is used to adapt bone-lengths of hand skeleton during training over different hand shapes, similar to the spirit in [1]. The predicted hand shape and pose parameters are fed to the hand model layer to infer hand shape $\mathbf{V}_h$ and its underlying joints $\mathbf{J}_h$ in the hand canonical coordinate system. Then we apply 3D transformation $\{s, \mathbf{R}, \mathbf{t}\}$ to hand shape $\mathbf{V}_h$ and hand joint $\mathbf{J}_h$, and get hand shape $\mathbf{V}$ and its binding joints $\mathbf{J}$ in the camera coordinate system.

In the second stage, we aims to refine the reconstructed hand shape in the first stage. Inspired by the fact that normalization is an effective approach to enhance hand pose estimation [22] and recognition task [37], we align the input point cloud by derotating the estimated joints on the palm in the first stage to a canonical pose, then feed to the second stage shape network to get the hand shape and hand joints in the hand canonical



(a) Detail of our network for hand shape recovery and hand pose estimation



(b) Detail of the shape network

Fig. 3. (a) Detail of our network for hand shape recovery and hand pose estimation. Our method consists of three stages, takes point cloud as input and generates the 3D hand shape of a hand and its 3D joints. Stage 1 and Stage 2 are named as hand shape reconstruction network, and Stage 3 is named as joint regression network. (b) Detail of the shape network.

coordinate system. Finally, we get the hand shape and joints to the camera coordinate using the inverse transformation of palm derotation.

For our shape network, we adopt the same network architecture used in [23], which is built upon PointNet++ [36]. As shown in Fig. 3(b), the shape network uses two units of grouping + per-point MLP, in which local features are extracted by nearest points grouping, multi-layer perception (MLP) and max-pooling. Finally, the hand pose θ, shape parameter β and hand joint locations $\mathbf{J}_{input}$ in the coordinate system of *input* point cloud are predicted by three parallel two-layer MLP units. For Stage 1, the coordinate systems of input point cloud is the camera coordinate system; For Stage 2, the coordinate systems of input point cloud is the aligned hand coordinate system with palm derotation.

B. Hand Model Layer

We represent a 3D hand shape model using SMPL model [12]. SMPL model is a statistical model for human shape, which encodes articulated human meshes with shape parameter and pose parameter. We implement SMPL model as a hand model layer (See Fig. 3(a)). The layer outputs user-specific hand shape $\mathbf{V}$ and joint locations $\mathbf{J}$ with given shape parameter β and joint angles θ.

Firstly, the template shape $\mathbf{T}(\beta, \theta)$ of a human hand at pose θ is encoded by a low dimensional linear subspace as follows

$$\mathbf{T}(\beta,\theta) = \mathbf{B}_0 + \sum_{n=1}^{|\beta|} \beta_n \mathbf{B}_n + \sum_{n=1}^{9K} (R_n(\theta) - R_n(\theta_0))\mathbf{P}_n \quad (1)$$

where the vector $\beta = [\beta_1, ..., \beta_{|\beta|}]^T$ is a shape parameter to learn, $\mathbf{B}_n \in R^{N\times 3}$ is a set of linear shape basis and

$\mathbf{B}_0 \in R^{N\times 3}$ is a mean hand shape at neutral pose θ_0. Denote $\{R_n(\theta)\}$ to be the n-th element of the concatenated vector $\{\mathbf{R}(\theta)\}$ with rotation matrices of all joints, and $\mathbf{P} = [\mathbf{P}_1, ..., \mathbf{P}_{9K}] \in R^{N\times 3\times 9K}$ to be the pose blend shape weight. The shape basis $\{\mathbf{B}_n\}_{n=1}^{|\beta|}$ and the mean hand shape $\mathbf{B}_0$ are estimated using principal component analysis (PCA) with the hand shapes at neutral pose θ_0 in our synthetic hand shape dataset.

Then we use a linear mapping matrix as [12] to get hand skeleton location $\mathbf{J}_0(\beta) \in R^{K\times 3}$ in neutral pose using a hand shape $\mathbf{T}(\beta, \theta_0)$ in neutral pose as follows

$$\mathbf{J}_0(\beta) = \mathbf{J}_{re}\mathbf{T}(\beta, \theta_0) \tag{2}$$

where the linear mapping matrix $\mathbf{J}_{re} \in R^{K\times N}$ is estimated with the hand shape and hand joints using the joint loss (See Section III D) on our synthetic hand shape dataset.

Secondly, we use linear blend skinning (LBS) to get the hand shape at a given hand pose θ. The hand shape with a given hand pose θ can be represented by the template hand shape $\mathbf{T}(\beta, \theta)$ and $\mathbf{J}_0(\beta)$ through the skinning weights $\mathbf{W} = \{w_{i,b}\}$ as follows

$$\mathbf{V}_i(\beta, \theta) = \sum_{b=1}^{K} w_{i,b}\mathbf{G}_b(\theta, \mathbf{J}_0(\beta))\mathbf{G}_b^{-1}(\theta_0, \mathbf{J}_0(\beta))\mathbf{T}_i(\beta, \theta) \tag{3}$$

where the skinning weight $w_{i,b}$ defines how the rotation of bone b affects a vertex $\mathbf{V}_i$. The matrix $\mathbf{G}_b(\theta, \mathbf{J})$ is a global transformation from local joint coordinate system to the world coordinate system, b_p is the parent index of bone b along the hand skeleton tree. The vector $\mathbf{T}_i(\beta, \theta)$ is the coordinate of vertex i in the template shape $\mathbf{T}(\beta, \theta)$ at pose θ.

We learn the skinning weight $\mathbf{W}$ and pose blend weight $\mathbf{P}$ with our synthetic hand shape dataset. The details can be found in the appendix.

C. Hand Pose Adaptation

The hand shape reconstruction network can be trained under full supervision on our synthetic hand shape dataset with ground truth shape. In order to make our model fully trainable on real data without hand shape supervision, we design a joint regression network and a Chamfer loss (See Section III-D) to bridge the domain gaps in hand pose.

We notice that the underlying hand skeleton joints of hand shape in our dataset are different to those in the public hand datasets (Fig. 2(b)). In order to handle the skeleton gaps in different datasets, we present a joint regression network for hand pose adaptation. The input of the joint regression network is a reconstructed hand shape via the hand shape reconstruction network, and the output is hand skeleton joints defined in each real dataset (NYU, ICVL, MSRA). Inspired by the spirit in [12] that human joints can be estimated with a linear mapping of human shape (similar to Eq.(2)), we design the joint regression network as a fully connected layer without nonlinear activation function. Our joint regression network can support more diverse and effective joint adaptation and achieve good generalization performance.

D. Loss Functions

Hand shape reconstruction network is trained with joint loss L_J, Chamfer loss L_C, shape loss L_S and mean shape loss L_{B_0}, and the shape loss L_S is used only if the ground truth 3D hand shape is available. The joint regression network is trained with joint loss L_J only. The total loss function for full network is defined as follows:

$$L_{total} = \lambda_J L_J + \lambda_C L_C + \lambda_{B_0} L_{B_0} + \mathbb{1}\lambda_S L_S \tag{4}$$

where $\lambda_J, \lambda_C, \lambda_{B_0}, \lambda_S$ are the loss weights, set to 2, 1, 1, 1, respectively, $\mathbb{1}$ is an indicator function that is 1 if ground truth 3D hand shape is available for a depth image and 0 otherwise.

Joint Loss We use Euclidean distance between the ground truth and predicted hand joints

$$L_J = \sum_{i=1}^{K} ||\mathbf{J}_i - \mathbf{J}_i^*||_2^2 \tag{5}$$

where $\mathbf{J}_i$ and $\mathbf{J}_i^*$ are the predicted and ground truth coordinates of joint i, K is the joint number. Since the skeleton joints in our synthetic dataset are different to those in real datasets (Fig. 2(b)), in the hand shape reconstruction network we use the common joints in our synthetic dataset and a real dataset to compute joint loss. In the joint regression network, we compute joint loss using joints commonly used for evaluation in a real dataset.

Chamfer Loss We use Chamfer distance to evaluate the distance from the input point cloud and predicted hand shape, which is designed to bridge the domain gaps in hand shape of synthetic and real data

$$L_C = \sum_{i=1}^{Q} ||\mathbf{p}_i - \mathbf{V}_{n(i)}||_2^2 \tag{6}$$

where $\mathbf{p}_i$ is i-th point of the input point cloud, and $\mathbf{V}_{n(i)}$ is the nearest vertex on the predicted hand shape to $\mathbf{p}_i$. In addition, we use a distance cut-off threshold $\tau = 5$ mm to ensure the loss function robust to the outliers, and Q is the number of points with $||\mathbf{p}_i - \mathbf{V}_{n(i)}|| \leq \tau$.

Shape Loss We compute Euclidean distance to evaluate the difference between the ground truth and predicted hand shape vertices. The shape loss is used when training and testing on our synthetic dataset, and we do not adopt the shape loss on real datasets.

$$L_S = \sum_{i=1}^{N} ||\mathbf{V}_i - \mathbf{V}_i^*||_2^2 \tag{7}$$

where $\mathbf{V}_i$ and $\mathbf{V}_i^*$ are the predicted and ground truth coordinates of vertex i of the hand shape, and $N = 1305$ is the number of vertices.

Mean Shape Loss We compute Euclidean distance between the mean hand shape $\mathbf{B}_0$ and user-specific template shape $\mathbf{T}(\beta, \theta)$, and enforce the shape variation of hand at neural pose to be small and reasonable

$$L_{B_0} = \sum_{i=1}^{N} ||\mathbf{T}_i(\beta, \theta_0) - \mathbf{B}_{0,i}||_2^2 \tag{8}$$

where $\mathbf{T}_i(\beta, \theta_0)$ and $\mathbf{B}_{0,i}$ are the coordinates of the i-th vertice in the predicted template shape and the mean shape at neural pose θ_0, respectively.

E. Implementation Details

The method is implemented with Tensorflow [38] with a NVIDIA Titan X GPU. We use ADAM optimizer to update weights with a batch size of 32. The learning rate is set to 0.001 and 0.9 weight decay per epochs. We train the whole network in an end-to-end manner, and we adopt the same loss for Stage 1 and Stage 2. Since the real hand datasets lack diversity of hand size or camera viewpoint, we conduct online data augmentation [18] by rotating, shifting and scaling the input hand point cloud.

1) Details of Data Augmentation: Since all the real benchmarks lack the diversity of hand shape or camera viewpoint, we conduct online data augmentation using rotation, shifting and scaling to make our model robust.

Rotation We apply a rotation on input point clouds and corresponding joint annotations. The rotation angles in Euler angles representation are sampled from uniform distribution of [-$\pi/6$, $\pi/6$], [-$\pi/6$, $\pi/6$], [-0, $\pi/2$], respectively.

Scaling Shape variance is limited in some datasets. We apply an in-plane scaling on input point clouds and corresponding joint annotations which add noise to x and y coordinates. The scaling factor sampled from uniform distribution of $[1/1.2, 1.2]$, $[1/1.2, 1.2]$ for x and y axis, respectively.

Shifting We add a random shifting noise to the input point clouds and joint annotations. The translation vectors are sampled from a normal distribution with a mean of 0 and a standard variance of 0.0125.

2) Details of Joint Supervisions: There are two kinds of joint supervisions in our network. The first one is at the end of joint regression network, and the second is at the end of each shape network. Since the joint regression network is designed to predict the desired hand joints of each datasets, we adopt the annotated hand joints in each dataset for joint supervision. In our shape network, we adopt joint loss supervision as one of weak supervisions for hand shape. However, as shown in Fig. 2(b), the predicted skeleton with our hand model layer has 21 joints, which are inconsistent to the hand skeletons in the main hand pose datasets. Due to the inconsistency, we perform approximated supervisions for the joints. The key idea is to use the nearest joint pair in our and annotated skeletons to compute the joint loss in the hand shape networks. The joints that do not have a proper approximated joints for supervision are not used to compute the joint loss of our shape network. During the training stage, we use equal weight for all joint loss functions at first, but drop joint loss of hand shape network by 10 times after training for 20 epochs, since it is only approximated supervision for the joints in hand shape network.

NYU dataset For NYU dataset, each frame has 36 joints annotation as shown in Fig. 4 (a) (left). As shown in Fig. 4 (b) (right), we choose finger joints with their indices in [31, 29, 27, 26, 25, 24, 22, 20, 19, 18, 16, 14, 13, 12, 10, 8, 7, 6, 4, 2, 1] of ground truth joints to compute the joint loss with our predicted joints. The rest of joints in the NYU dataset are not used for joint supervision of the hand shape network because they are so far from the hand joints in our hand model (See Synthetic Data of Fig. 2(b)) to be a good supervision. The approximated correspondences between the hand joints of our hand model (blue circles) and the annotated joints for supervision (red circles) are connected with blue lines in Fig. 4 (a) (right).

ICVL dataset For ICVL dataset, we first transform our joints by adding a virtual palm center joint as the middle point of our wrist center (indexed as 1) and our root joint of middle finger (indexed as 10), then choose finger joints with their indices in [1, 3, 4, 5, 7, 8, 9, 11, 12, 13, 15, 16, 17, 19, 20, 21] from our transformed joints shown in Fig. 4 (b) (right) to compute the joint loss. The pairs of our joints of our model (blue circles) and the annotated hand joints for supervision (red circles) are connected with blue lines in Fig. 4 (b) (right).

IV. Experiments

A. Datasets and Evaluation Metrics

NYU dataset contains 72K training frames and 8K testing frames with 3D hand pose annotations. Each frame has 36 annotated joints, we choose 21 joints close to our predefined skeleton joints to supervise the joints of hand shape network, and we adopt the commonly used 14 joints [3] for hand pose evaluation.

ICVL dataset contains 16K training frames and 1596 testing frames. Each frame has 16 annotated joints, which are used to supervise part of joints in hand shape network.

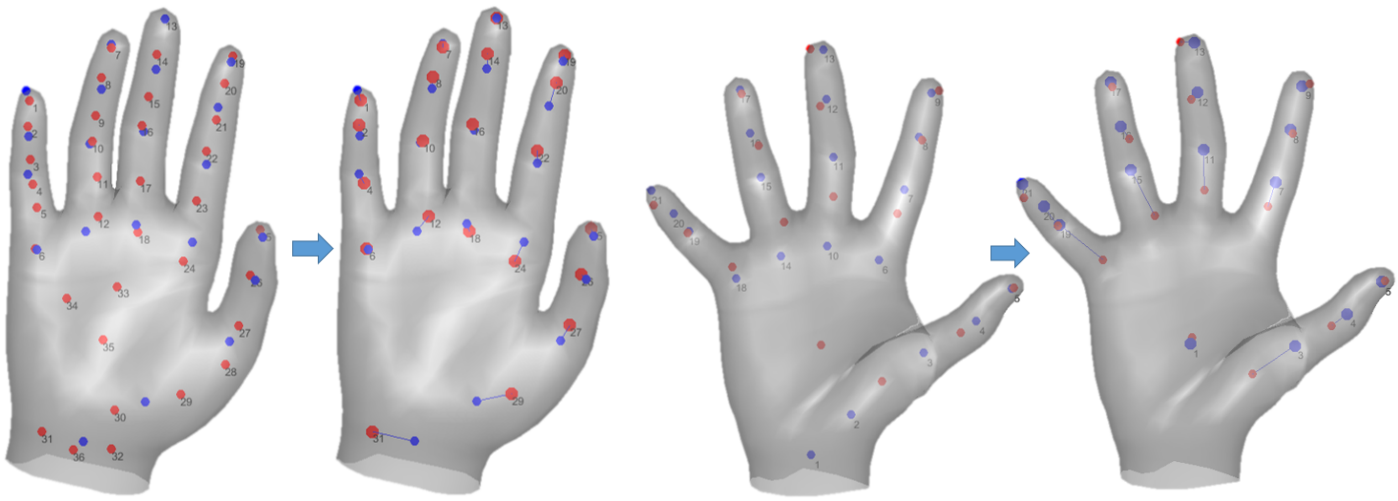
MSRA dataset contains 76K frames which captured from 9 different subjects. We follow the protocol [18] and use leave-one-out cross validations for evaluation.

Our synthetic hand shape dataset consists of 270K synthetic depth image with hand shape and hand pose annotation, 243K frames for training and 27K frames for testing. This dataset is mainly used to evaluate the hand shape performance of our method in the ablation study.

To evaluate hand pose, we use two standard evaluation metrics, which are widely used in many hand pose estimation work [39], [40], [41], average joint error and percentage of good frames where maximum joint error under a given threshold. To evaluate hand shape, on our synthetic hand shape data we use the mean error between the mesh vertices of ground truth and predicted hand shape, named shape error; On real datasets we use the mean Chamfer distance from input point cloud to the predicted vertices of hand shape, named point cloud error, due to the lack of ground truth shapes. We also use qualitative results for evaluation.

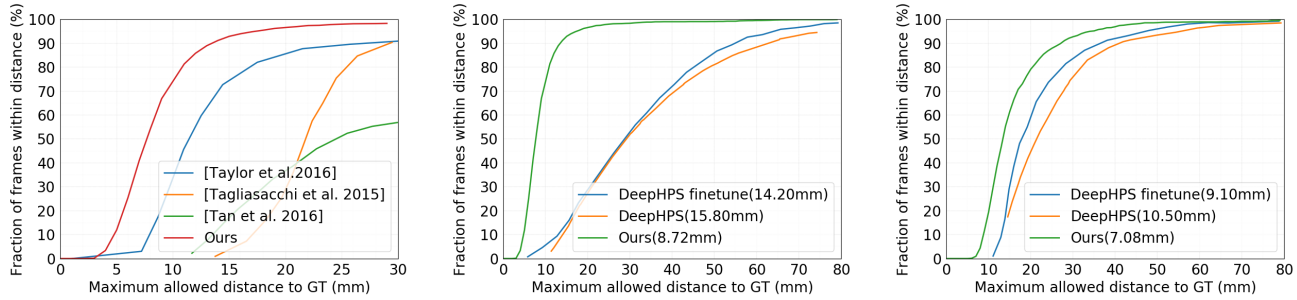
B. Comparison to State-of-the-art Methods

The public hand datasets do not contain hand shape annotations, thus we compare to the state-of-the-art hand shape recovery methods, DeepHPS [1], Tagliasacchi *et al.* [42], Tan *et al.* [10], Taylor *et al.* [27] using the popular evaluation metrics, such as joint error in previous hand shape methods [10], [1]. DeepHPS [1] is the state-of-the-art CNN-based hand shape recovery method from depth. The methods [42], [10],



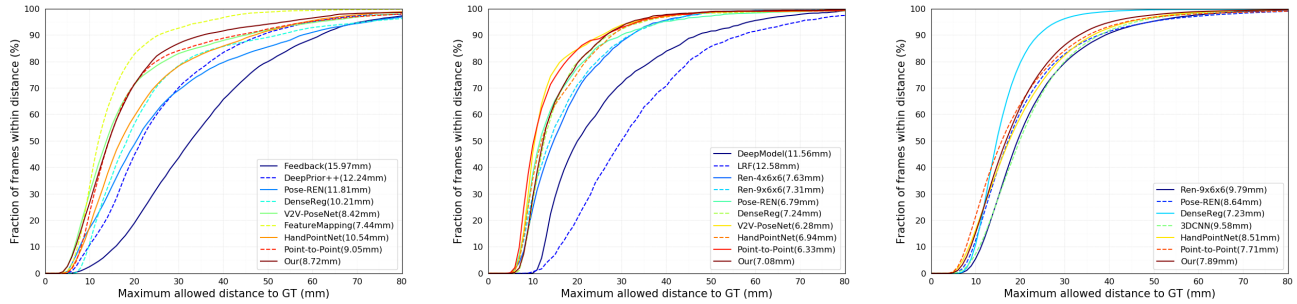
(a) Annotation and our skeleton on NYU dataset. (b) Annotation and our skeleton on ICVL dataset.

Fig. 4. Details of joint supervision. The red solid circles denote joints of the ground truth annotation, and the blue solid circles denote joints of our skeleton. By selection and transformation, the pairs of joints used to compute joint loss are connected with blue lines.



(a) Ours vs. [42][27][10] on NYU (b) Ours vs. [1] on NYU (c) Ours vs. [1] on ICVL

Fig. 5. Comparisons with hand shape recovery methods on NYU and ICVL datasets.



(a) NYU (b) ICVL (c) MSRA

Fig. 6. Comparisons with hand pose estimation methods on NYU, ICVL and MSRA.

[27] are all model-based methods using nonlinear optimization, which need several frames to initialize personalized hand shape model and use temporal information to estimate hand shape and pose. For fair comparison, we train our network from scratch on all datasets.

The methods [42], [27], [10] conduct evaluations only on the first 2,440 test frames in NYU dataset, and use ten joint positions to compute error in each frame. In order to conduct fair comparison with them, we report results for the same test frames and the same joints. During the comparison with [1], we adopt the whole test set and the joints used in [22] for evaluation. Fig. 5 (a) shows the comparison with the hand shape recovery methods [42], [27], [10] on NYU dataset. We observe that our method outperforms [42], [10], [27] on NYU dataset. Fig. 5 (b) compares with the hand shape recovery method DeepHPS [1] on NYU, and Fig. 5 (c) compares with DeepHPS [1] on ICVL. DeepHPS and DeepHPS:finetune are the models trained from scratch and fine-tuned from the synthetic datasets in [1], respectively. We observe that our method is superior to DeepHPS [1] for both

TABLE I
3D JOINT ERROR (IN MM) USING DIFFERENT BACKBONE NETWORKS ON NYU AND ICVL DATASETS.

Network	NYU	ICVL
Ours (2DCNN, one-stage)	13.96	9.94
Ours (PointNet++, one-stage)	13.70	8.67
DeepHPS [1]	15.80	10.50

datasets. The mean joint errors of our method are 7.08mm and 5.48mm lower than those of DeepHPS and DeepHPS finetuned on NYU dataset, 2.90mm and 1.40mm lower than those of DeepHPS and DeepHPS finetuned on ICVL dataset. The point cloud error from input point cloud to the predicted hand shapes with our method and DeepHPS are 5.44mm and 10.67mm, respectively. Therefore, hand shape models with our method get better alignment with the input point cloud. We compare our method with a recent method [2] that is the current state-of-the-art single depth based hand shape recovery method, and we find that our method also achieves better hand pose estimation on NYU dataset (Ours: 8.72mm, [2]:11.8mm).

We also compare the performance of our method with the hand pose estimation methods on NYU, MSRA, ICVL datasets (refer to Fig. 6). All the compared methods here can not predict the hand shape. Our method achieves competitive performance on three datasets, only 0.33mm, 0.75mm, 0.18mm worse than state-of-the-art methods on NYU, ICVL and MSRA dataset. The performance on ICVL and MSRA dataset may subject to randomness given the noisy ground truth as mentioned in [43].

Our method achieves better performance than DeepHPS [1]. In order to investigate whether the main performance gain is due to the used different backbone networks in our network and DeepHPS (Ours:PointNet++, DeepHPS:2DCNN), we design almost the same network to DeepHPS [1], named 'Ours (2DCNN, one-stage)', in which Region Ensemble (REN) [44] is used as the backbone network of PoseCNN [1] and the same ShapeCNN as [1] is adopted to estimate the hand shape, and then we use Chamfer loss to supervised the predicted hand shape by ShapeCNN and put our joint regression network after the recovered hand shape. For fair comparison, our method and DeepHPS use the same training data, and do not pretrain on synthetic datasets. From Table I, we can see the 3D joint error of our method is better than those of DeepHPS for NYU dataset and ICVL dataset though the same backbone network and the similar network architecture are used in DeepHPS and 'Ours (2DCNN,one-stage)'. Therefore, the weak supervision of hand shape with Chamfer loss and the joint regression network are a key source of our main performance gain. Then, in order to verify that our main performance gain over DeepHPS is not due to the used backbone network PointNet++, we conduct comparison experiments with 'Ours (2DCNN,one-stage)' and a PointNet++ based one-stage network by removing the shape network in Stage 2 from our full model (See Fig. 3(a)), which is named 'Ours (PointNet++,one-stage)'. From the comparison between 'Ours (2DCNN,one-stage)' and 'Ours (PointNet++,one-stage)', we observe that PointNet++ is effective to improve the performance of network, but it does not lead to significant performance gain.

C. Ablation Study

We use NYU dataset [3] and our synthetic dataset to evaluate the effect of each component in our model. The joint numbers of the hand joint regression network (denoted as reg-joint) on NYU dataset and the underlying joints of our hand shape model are 14 and 21, respectively. To conduct fair comparison with/without joint regression, we set the joint number of reg-joint as 21 in the ablation study.

Effect of Chamfer Loss. We firstly evaluate the effect of Chamfer loss on hand shape and hand pose estimation using our synthetic dataset. Table II (a) compares the joint loss, joint + Chamfer loss and joint + shape loss. Though the mean joint errors are similar for the three types of loss functions, the shape error and point cloud error reduce 2.26mm and 0.57mm by adding Chamfer loss.

The shape error with Chamfer + joint loss (i.e. under weak supervision) is only 0.48mm higher than that with fully supervised shape + joint loss. We also show the impact of Chamfer loss on joint error using NYU dataset in Table II (b). The joint error without Chamfer loss is 0.48mm higher than our full model with Chamfer loss. Fig. 7 shows the impact of Chamfer loss on the hand shape recovery qualitatively. Though the joint loss can solely enforce good hand pose estimation, Chamfer loss can improve the alignment between the recovered hand shape and the hand point cloud as well as the hand pose. Therefore, Chamfer loss can enhance the hand shape recovery and the hand pose estimation results.

Effect of Joint Regression Network. We evaluate the effect of our joint regression network on hand pose estimation using NYU dataset. Table II (b) compares the hand joint error and point cloud error with and without the joint regression network. The joint error and point cloud error are reduced by 2.89mm and 0.46mm using the joint regression network. Therefore, our joint regression network can reduce the gap between our predefined hand skeleton and the hand skeleton in the real dataset.

Effect of Shape Network in Stage 2. We evaluate the effect of shape network in Stage 2 using NYU dataset by removing it. From Table II (b), we can see the 3D joint error and the point cloud error increase by about 5.0 mm and 2.2 mm after removing the shape network in Stage 2 while keeping the rest of our network remained. Therefore, shape network in Stage 2 benefits the hand shape recovery and hand pose estimation.

Effect of Mean Shape Loss. We evaluate the effect of mean shape loss on hand shape and hand pose estimation using NYU dataset. As we can be seen in Table II(b), removing mean shape loss increases the 3D joint error and point cloud error by 0.6mm and 1.22mm. These results show that mean shape loss contributes to the final performance. As shown in Fig. 8, removing mean shape loss may lead to unnatural finger shape deformations in the recovered hand shape (row 2 to row 4), and also the misalignment between the recovered hand shape and the input point cloud (row 1).

D. Qualitative Results

Fig. 9 shows qualitative results of hand shape reconstruction and pose estimation on NYU, MSRA, ICVL datasets. We

TABLE II
ABLATION STUDY (IN MM). (A) EFFECT OF DIFFERENT LOSS FUNCTIONS ON SHAPE RECONSTRUCTION AND HAND POSE ESTIMATION ON OUR SYNTHETIC DATASET. (B) EFFECT OF JOINT REGRESSION NETWORK (REG-JOINT), CHAMFER LOSS, THE SHAPE NETWORK IN STAGE 2 (SHAPE-NET 2) AND MEAN SHAPE LOSS (MEAN-SHAPE) ON NYU DATASET.

Loss	3D Joint Error	Shape Error
joint	3.12	6.46
joint + Chamfer loss	3.12	4.2
joint + Shape loss	3.13	3.72

(a) Effect of different loss functions

Network	w/o reg-joint	w/o Chamfer loss	w/o shape-net 2	w/o mean-shape	full
3D Joint Error	11.61	8.38	13.13	8.67	8.20
Point Cloud Error	5.90	6.16	7.56	6.63	5.39

(b) Effect of different network modules and loss functions

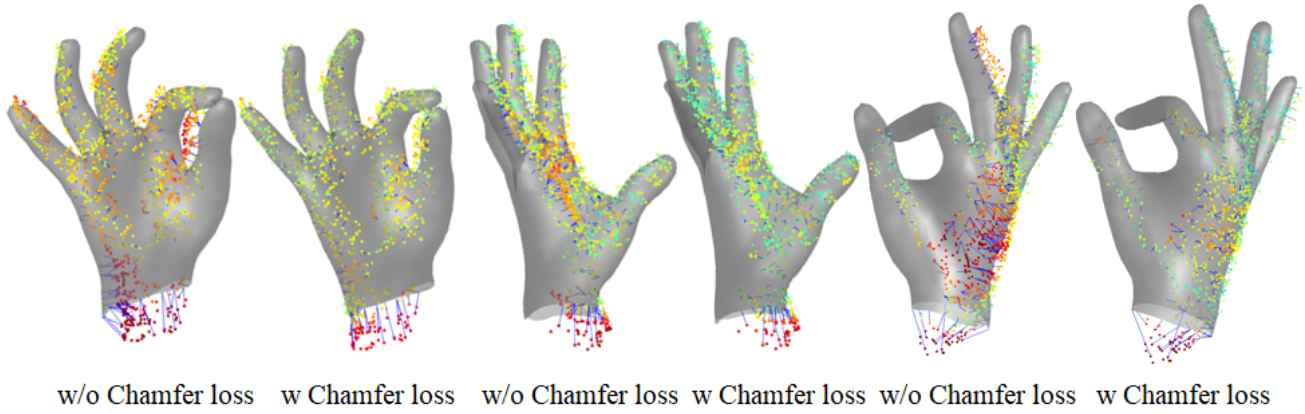


Fig. 7. Comparisons of the recovered hand shape without/with Chamfer loss in NYU dataset. We show the correspondences between recovered hand shape and input point cloud. Chamfer loss helps to align the recovered hand shape with the point cloud better.

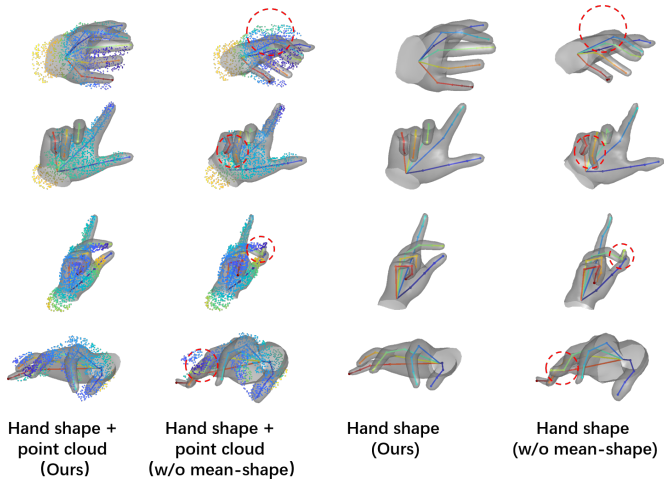


Fig. 8. Comparisons of the recovered hand shape without/with mean shape loss (mean-shape) in NYU dataset. The hand shape reconstruction errors without mean-shape are highlighted with red circles.

show the hand joints with the joint regression network and the reconstructed hand shape models with the underlying joints, and we compare our hand shape models with those with DeepHPS [1] on NYU dataset. We can observe that our method can achieve accurate 3D hand shape and pose results, our shape models get better alignments with the input point cloud than those with [1], and the joint regression network can adapt our hand skeleton model to different skeleton models in real datasets.

We also provide more qualitative results for NYU, ICVL, MSRA and our synthetic datasets in Fig. 10, Fig. 11, Fig. 12 and Fig. 13, respectively. We can see that the recovered hand shape models with our method get better alignment with the input point cloud than those with [1] on NYU dataset. From Fig 10 to Fig. 13, we observe that the hand shape and hand joints are accurate under a variety of hand gestures.

V. CONCLUSION

In this paper, we present a deep learning framework to recovery hand shape from the point cloud of a single depth image, which can be trained in a weakly-supervised manner on real data using only easily accessible hand skeleton pose annotations. We design a joint regression network that facilitates the hand pose adaptation of different hand skeleton definitions. We also introduce a Chamfer loss to encourage geometric consistency between our prediction and the point cloud of input depth. We find that the Chamfer loss can enhance the hand shape performance in a weakly supervised manner. With the help of these specific design for domain adaptation, our model learns both hand pose and shape knowledge from the training data at the same time, and outperforms state-of-art hand shape recovery methods. Experiments demonstrate that the our method achieves better performance than the state-of-art hand shape recovery methods. Although our method is designed for hand shape recovery, it can also inspire related researches such as human body and face shape reconstruction.

APPENDIX

ESTIMATION OF SKINNING WEIGHT AND POSE BLEND WEIGHT

We follow [12] to learn the skinning weight $\mathbf{W}$ and pose blend weight $\mathbf{P}$ with our synthetic hand shape dataset. To this

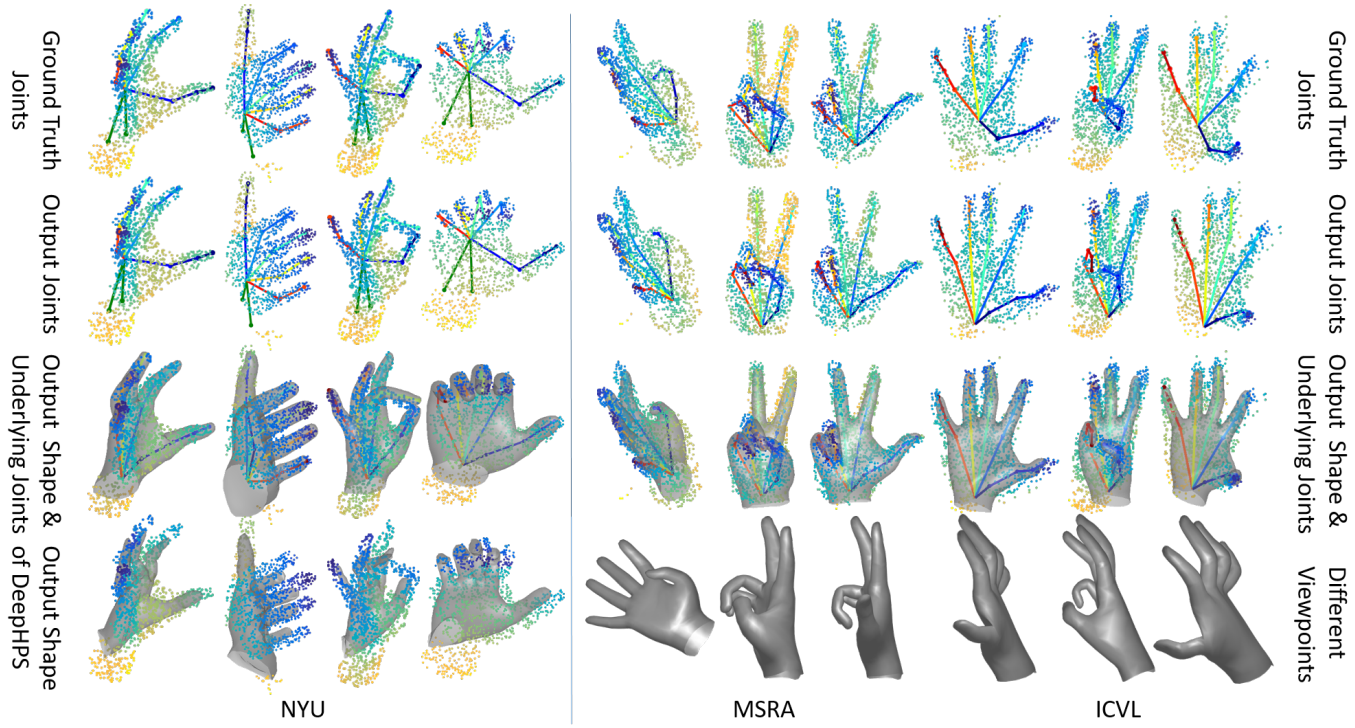


Fig. 9. Qualitative results for NYU [3], MSRA [39] and ICVL [8] datasets. We compare the output hand shape with our method and DeepHPS [1], as well as output joints with our method and ground truth joints.

end, we minimize the vertex reconstruction error using the following object function

$$L(\mathbf{W},\mathbf{P}) = \sum_{s}\sum_{p}\sum_{n} ||\mathbf{V}^{*}_{s,p,i} - \mathbf{V}_{s,p,i}(\mathbf{W},\mathbf{P},\beta_s,\theta_{s,p})|| \tag{9}$$

where $\mathbf{V}^{*}_{s,p,i}$ is the ground truth position of the i-th vertex of subject s at the pose p, $\mathbf{V}_{s,p,i}(\mathbf{W},\mathbf{P},\beta_s,\theta_{s,p})$ is the estimated position of the i-th vertex of subject s at the pose p with the shape parameter β_s and pose parameter $\theta_{s,p}$ via Eq. (3).

In Eq. (9), the hand shape parameter β_s can be obtained by projecting the hand shape $\mathbf{T}(\beta_s,\theta_0)$ at the neutral pose to the orthogonal basis $\{\mathbf{B}_n\}_{n=1}^{|\beta|}$, and the hand shape $\mathbf{T}(\beta_s,\theta_0)$ and the hand joints $\mathbf{J}_0(\beta_s))$ required by $\mathbf{V}_{s,p,i}$ can be both exported during when we construct our synthetic hand shape dataset using Maya.

The above optimization problem is solved with Tensorflow using ADAM optimizer. The learning rate is set to 0.01 and 0.9 weight decay per epochs. In the real training dataset, the skinning weight and pose blend weight are not optimized.

ACKNOWLEDGMENT

The authors are grateful to the editors and reviewers for their careful review and inspiring suggestions. The authors would like to thank Dr. Jameel Malik for helpful discussions and help.

REFERENCES

[1] J. Malik, A. Elhayek, F. Nunnari, K. Varanasi, K. Tamaddon, A. Heloir, and D. Stricker, "Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 110–119.

[2] J. Malik, A. Elhayek, F. Nunnari, and D. Stricker, "Simple and effective deep hand shape and pose regression from a single depth image," *Computers & Graphics*, pp. 85–91, 2019.

[3] 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)*, vol. 33, no. 5, p. 169, 2014.

[4] G. Moon, J. Yong Chang, and K. Mu Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5079–5088.

[5] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Dual grid net: hand mesh vertex regression from single depth maps," in *Proceedings of the European Conference on Computer Vision*, 2020.

[6] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3d pose inference from synthetic images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4663–4672.

[7] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2107–2116.

[8] 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*, 2014, pp. 3786–3793.

[9] S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, and A. Fitzgibbon, "Learning an efficient model of hand shape variation from depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2540–2548.

[10] D. Joseph Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton, "Fits like a glove: Rapid and reliable hand shape personalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5610–5619.

[11] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 245, 2017.

[12] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, p. 248, 2015.

[13] A. Tkach, M. Pauly, and A. Tagliasacchi, "Sphere-meshes for real-time

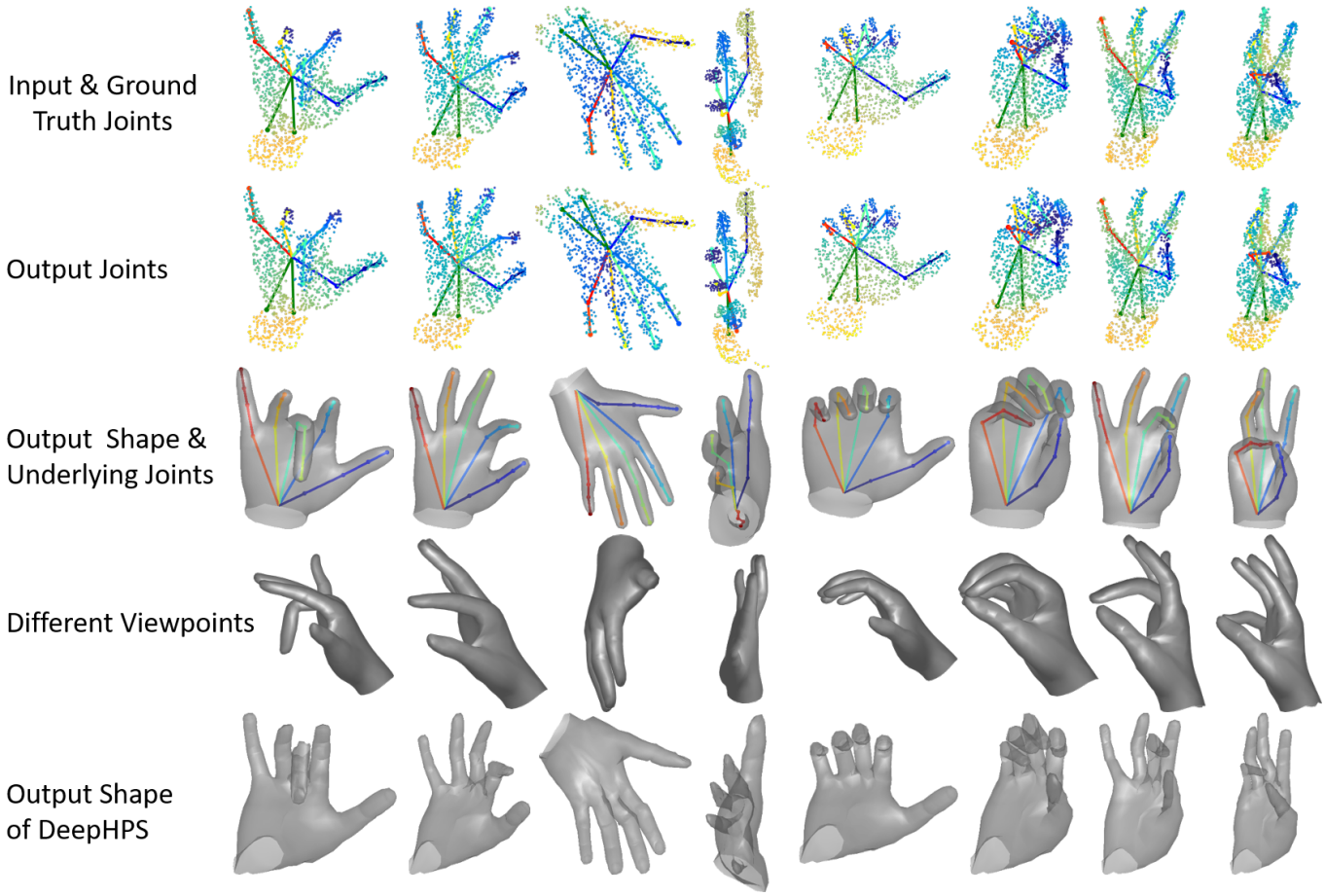


Fig. 10. Qualitative results comparing to DeepHPS [1] on NYU dataset. We visualize the output hand shape with underlying joints, and a rendered image under a different viewpoint, as well as output joints and ground truth joints.

hand modeling and tracking," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 222, 2016.

[14] 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*, vol. 55, pp. 404–414, 2018.

[15] 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.

[16] X. Chen, G. Wang, C. Zhang, T.-K. Kim, and X. Ji, "Shpr-net: Deep semantic hand pose regression from point clouds," *IEEE Access*, vol. 6, pp. 43 425–43 439, 2018.

[17] 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, pp. 1991–2000.

[18] M. Oberweger and V. Lepetit, "Deepprior++: Improving fast and accurate 3d hand pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 585–594.

[19] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dense 3d regression for hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5147–5156.

[20] 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*, 2016, pp. 3593–3601.

[21] 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.

[22] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand pointnet: 3d hand pose estimation using point sets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[23] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3d hand pose estimation," *Proceedings of European Conference on Computer Vision*, 2018.

[24] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.

[25] X. Gao, W. Hu, and G.-J. Qi, "GraphTER: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[26] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon, "User-specific hand modeling from monocular depth sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 644–651.

[27] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 143, 2016.

[28] C.-L. Li, T. Simon, J. Saragih, B. Poczos, and Y. Sheikh, "Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3d hand shape and pose estimation from a single rgb image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[30] Artec, "Artec eva," https://www.artec3d.com.

[31] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proceedings of the International Conference on Computer Vision*, 2017.

[32] Manu, "Non-rigid icp," 2019, https://www.mathworks.com/matlabcentral/fileexchange/4 nonrigidicp.

[33] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "Bighand2.

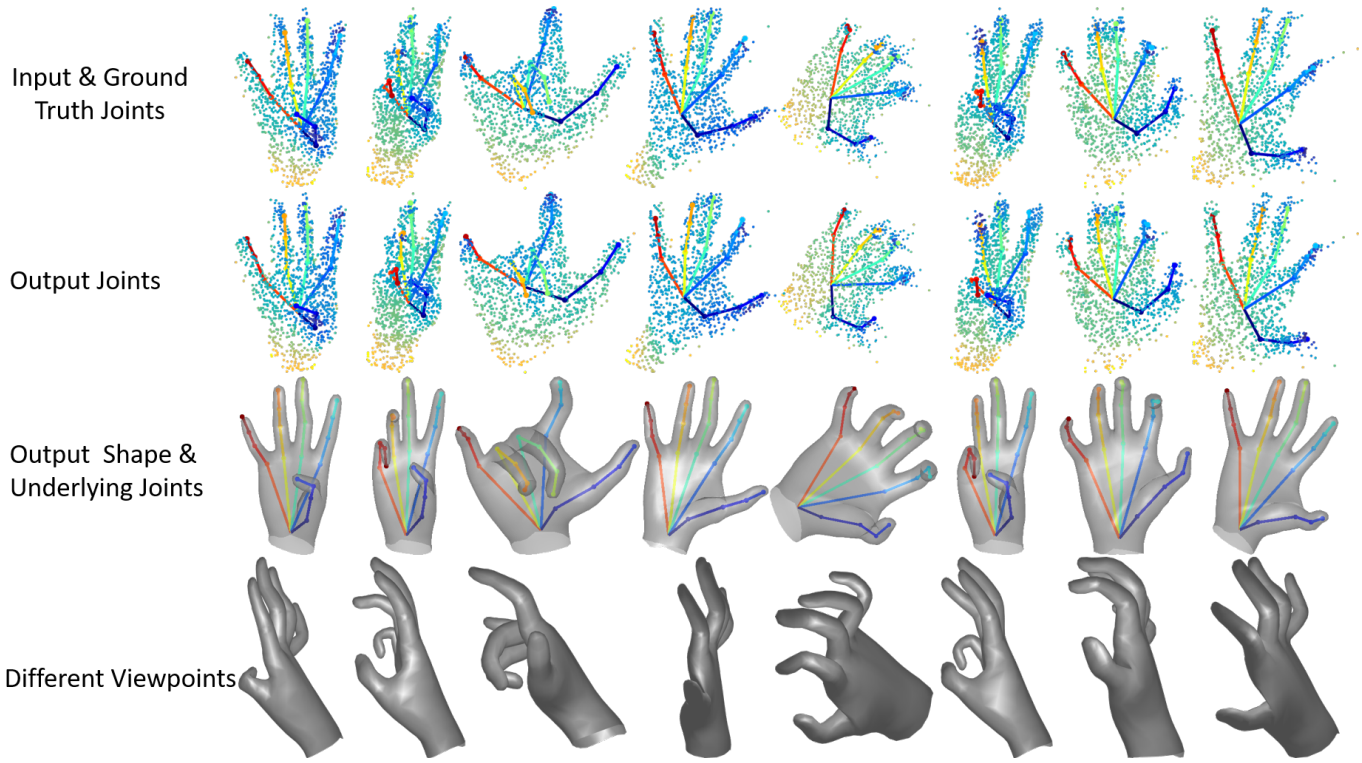


Fig. 11. Qualitative results on ICVL dataset. We visualize the output hand shape with underlying joints, and a rendered image under a different viewpoint, as well as output joints and ground truth joints.

2m benchmark: Hand pose dataset and state of the art analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4866–4874.

[34] Autodesk, "Maya," https://area.autodesk.com/.

[35] W. Jakob, "Mitsuba renderer," 2010, http://www.mitsuba-renderer.org.

[36] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[37] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[38] S. S. Girija, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *Software available from tensorflow. org*, 2016.

[39] 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*, 2015, pp. 824–832.

[40] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands Deep in Deep Learning for Hand Pose Estimation," in *CVWW*, 2015.

[41] J. S. Supancic III, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: methods, data, and challenges," *ICCV*, 2015.

[42] 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*, vol. 34, no. 5. Wiley Online Library, 2015, pp. 101–114.

[43] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3d training data for fine hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4957–4965.

[44] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4512–4516.

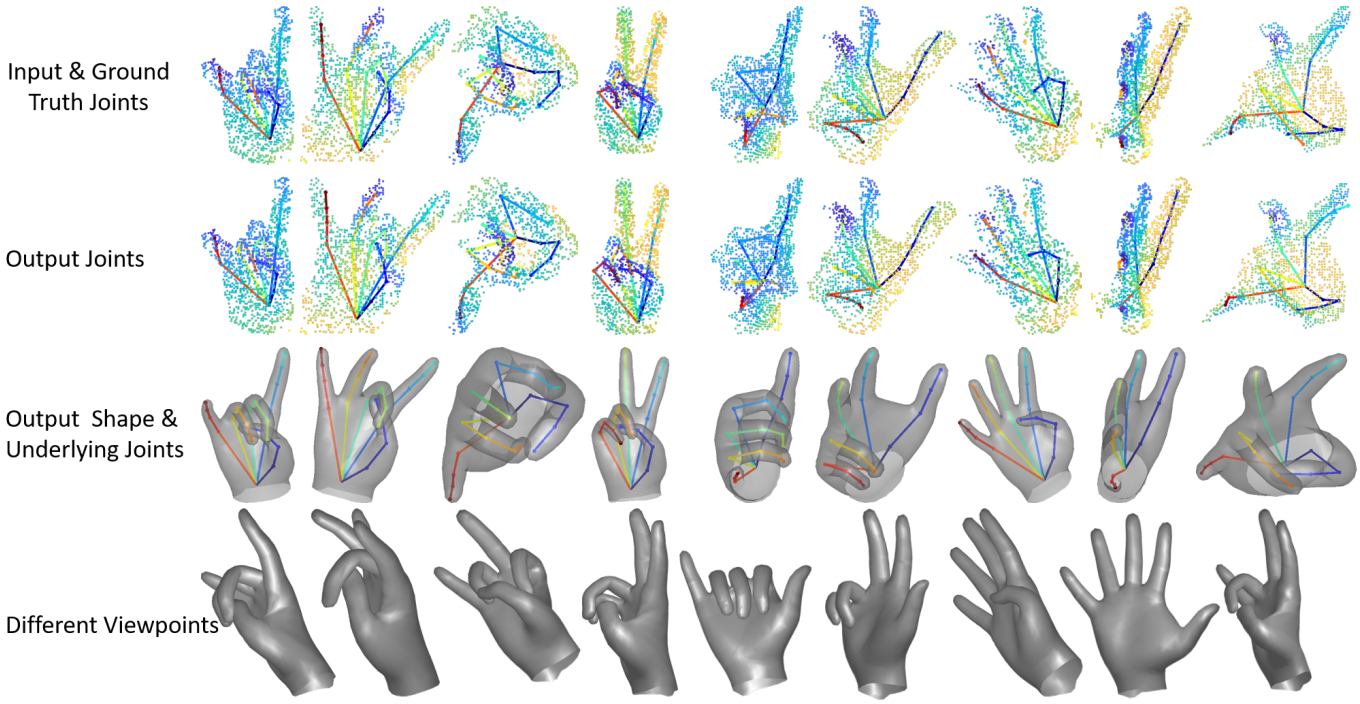


Fig. 12. Qualitative results on MSRA dataset. We visualize the output hand shape with underlying joints, and a rendered image under a different viewpoint, as well as output joints and ground truth joints.

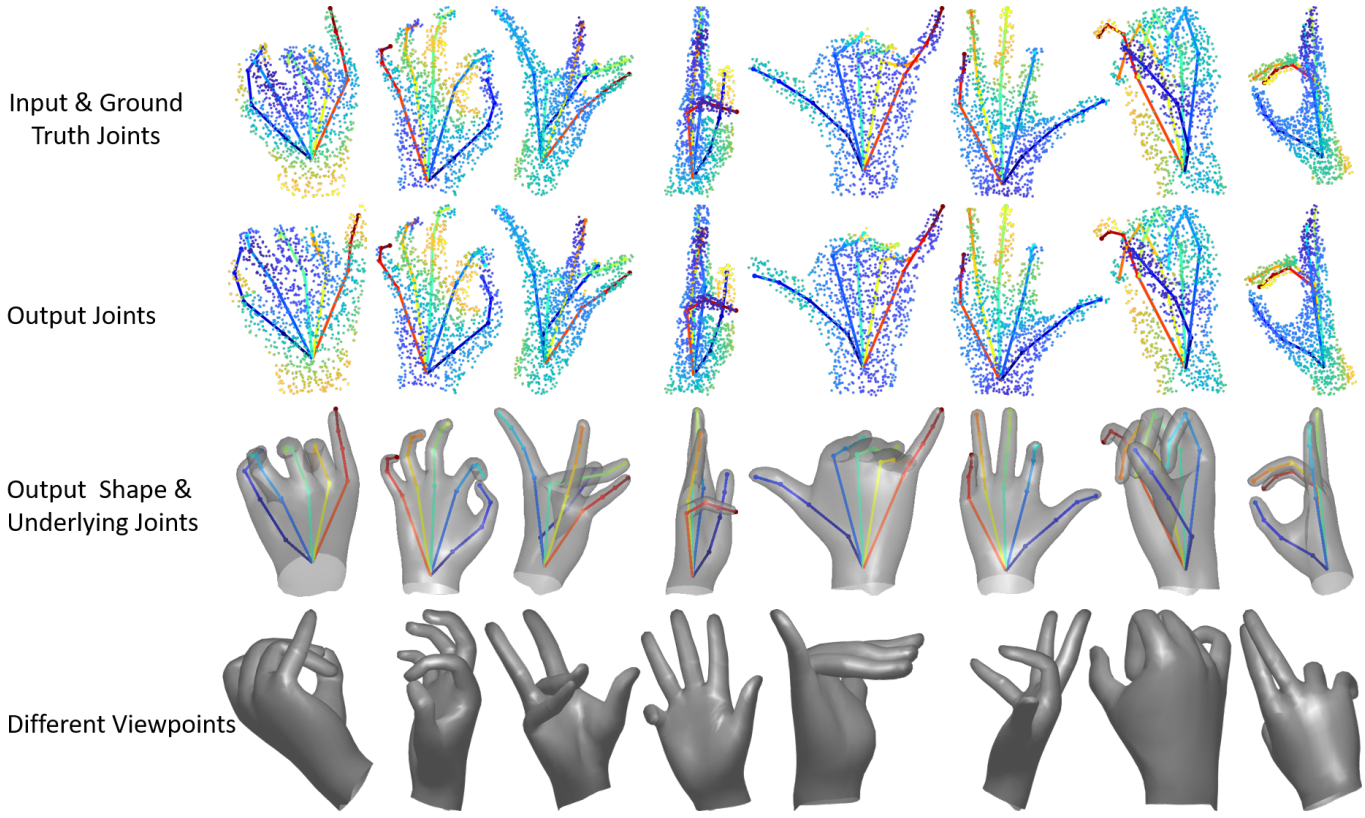


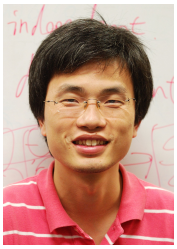
Fig. 13. Qualitative results on our synthetic dataset. We visualize the output shape with underlying joints, and a rendered image under a different viewpoint, as well as output joints and ground truth joints.

Xiaoming Deng received the BS and MS degrees from Wuhan University, in 2001 and 2004, respectively, and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (CAS), in 2008. After a two-year postdoctoral at Institute of Computing Technology, CAS, he joined Institute of Software, CAS, in 2010, and he is currently an Associate Professor. He has been a Research Fellow at National University of Singapore from 2012 to 2013. He is main research topics are in computer vision, and specifically related to 3D reconstruction, human motion tracking and synthesis, and deep learning.

Yuying Zhu is a Master graduate student at the Institute of Software, Chinese Academy of Sciences. Before that, she received a Bachelor degree from University of Science and Technology of China, in 2016. Her main research interest is computer vision and human-computer interaction.

Yinda Zhang is Research Scientist at Google. He received the Ph.D. degree from Princeton University, advised by Professor Thomas Funkhouser. Before that, he received a Bachelor degree from Department of Automation, Tsinghua University and a Master degree from Department of Electrical and Computer Engineering, National University of Singapore under the supervision of Professor Ping Tan and Professor Shuicheng Yan. He is currently working on 3D context model, 3D deep learning, and scene understanding.

Zhaopeng Cui received the Ph.D. degree from Simon Fraser University in 2017. He was a senior researcher at ETH Zurich. He is currently a research professor in the College of Computer Science, Zhejiang University. His research interests include 3D mapping and localization, 3D scene understanding, image and video editing.



Ping Tan is currently an Associate Professor with the School of Computing Science of the Simon Fraser University in Canada. Before that, he was an Associate Professor with the National University of Singapore. He received the Ph.D. degree from the Hong Kong University of Science and Technology in 2007, and the bachelors and masters degrees from Shanghai Jiao Tong University in China in 2000 and 2003, respectively. His research interests include computer vision, computer graphics, and robotics. He has served as an Editorial Board Member of the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), the International Journal of Computer Vision (IJCV), Computer Graphics Forum (CGF), and the Machine Vision and Applications (MVA). He served as an Area Chair for CVPR, SIGGRAPH, SIGGRAPH Asia, and IROS.

Wentian Qu is a Master graduate student at the Institute of Software, Chinese Academy of Sciences. Before that, he received a Bachelor degree from China University of Geosciences in Beijing, in 2018. His main research interest is computer vision and human-computer interaction.

Cuixia Ma received the B.S. and M.S. degrees from Shandong University, China, in 1997 and 2000, respectively, and the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, in 2003. She was a Research Associate in the Department of Computer Science, Naval Postgraduate School in Monterey, CA, USA, from 2005 to 2006. She is a Professor with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, and Beijing Key Laboratory of Human-Computer Interaction. Her research interests include human computer interaction and multimedia computing.

Hongan Wang received the PhD degree in computer science from Institute of Software, Chinese Academy of Sciences. He is a Professor and the Director of Beijing Key Laboratory of Human-Computer Interactions Laboratory at Institute of Software, Chinese Academy of Sciences. His research interest is real-time reasoning and intelligent user interaction.