IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 7, 2018, accepted August 1, 2018, date of publication August 6, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2863540

SHPR-Net: Deep Semantic Hand Pose Regression From Point Clouds

XINGHAO CHEN[1], GUIJIN WANG[1], (Senior Member, IEEE), CAIRONG ZHANG[1], TAE-KYUN KIM[2], (Member, IEEE), AND XIANGYANG JI[3]

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K.
[3]Department of Automation, Tsinghua University, Beijing 100084, China

Corresponding author: Guijin Wang (wangguijin@tsinghua.edu.cn)

ABSTRACT 3-D hand pose estimation is an essential problem for human–computer interaction. Most of the existing depth-based hand pose estimation methods consume 2-D depth map or 3-D volume via 2-D/3-D convolutional neural networks. In this paper, we propose a deep semantic hand pose regression network (SHPR-Net) for hand pose estimation from point sets, which consists of two subnetworks: a semantic segmentation subnetwork and a hand pose regression subnetwork. The semantic segmentation network assigns semantic labels for each point in the point set. The pose regression network integrates the semantic priors with both input and late fusion strategy and regresses the final hand pose. Two transformation matrices are learned from the point set and applied to transform the input point cloud and inversely transform the output pose, respectively, which makes the SHPR-Net more robust to geometric transformations. Experiments on NYU, ICVL, and MSRA hand pose data sets demonstrate that our SHPR-Net achieves high performance on par with the start-of-the-art methods. We also show that our method can be naturally extended to hand pose estimation from the multi-view depth data and achieves further improvement on the NYU data set.

INDEX TERMS Human computer interaction, hand pose estimation, deep learning, machine learning, point cloud.

I. INTRODUCTION

Fast and accurate 3D hand pose estimation is an essential technique for human computer interaction and virtual/augmented reality [1], since it provides foundational skeleton information for hand gesture recognition [2], [3] and hand interaction [4]. As the wide availability of commercial depth cameras such as Microsoft Kinect [5] and Intel Realsense [6], depth-based hand pose estimation has attracted much research interest in the last decade [7]–[22].

Depth-based hand pose estimation has advanced significantly recently, especially due to the successful application of deep learning. Most existing methods [7], [8], [10]–[13] treat depth maps as images with one channel and feed them into a 2D convolutional neural networks (CNN). However, mapping a 2D depth image to 3D joint coordinates is a highly challenging learning task due to the disparate domains of input and output. Recent studies [9], [19], [23] convert depth images into volumetric representations and utilize 3D CNNs to estimate hand pose. This tends to be more efficient to capture the geometric properties of depth maps and eases the burden of network learning. Yuan *et al.* [22] obtained the observations that 3D representations outperform 2D depth maps by comparing tens of state-of-the-art methods in Hands In the Million (HIM 2017) challenge [24]. However, 3D volumetric representations bring potential quantization artifacts and require large memory for high voxel resolution.

Point cloud is a set of points represented by coordinates and it is more simple yet effective representation for depth data. In this manner, the input point set and the output hand pose share the same domain of representation, which can benefit the learning of mapping input data to output pose. Driven by latest PointNet [25] and PointNet++ [26] that achieve impressive performance on object classification and semantic segmentation tasks, Ge *et al.* [21] proposed a method to directly predicting hand poses from point sets via hierarchical PointNets and showed promising performance for the problem of hand pose estimation. However, there are still more challenges to be tackled. Hand PointNet [21] is not totally

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

an end-to-end network as it addresses the hand orientations by normalizing the input point cloud via principle component analysis (PCA) and refines fingertip using an additional post-processing network. Additionally, PointNet/PointNet++ is a generic architecture for 3D deep learning and does not fully exploit the relations between input points and hand joints, which is highly desired in the problem of hand pose estimation.

In this paper, we propose an end-to-end deep Semantic Hand Pose Regression network (SHPR-Net) for estimating hand poses from point clouds. The SHPR-Net takes a point cloud as input and predicts the corresponding hand pose. It consists of two subnetworks. The semantic segmentation subnetwork (SegNet) performs point-wise classification to segment the point cloud into different semantic parts. The original point cloud, together with the semantic segmentation representations, are fed into a pose regression subnetwork (RegNet) to regress the hand pose. Semantic information is also fused into the RegNet in the last fully connected (fc) layer to enhance the performance of hand pose regression. Two transformation matrices are learned from the input point cloud using mini-PointNets (T-Nets) to transform the input points and inversely transform the output of the RegNet respectively. In this manner, our SHPR-Net is more robust to geometric transformations of input clouds and tends to ease the burden of network learning. To make sure the output transformation matrix is the inverse matrix of the input one, the identity matrix loss is introduced to encourage the matrix multiplication of these two transformation matrices is equal to an identity matrix. Experiments on three challenging depth-based hand pose datasets (ICVL, MSRA, NYU) demonstrate that our SHPR-Net can achieve strong performance on par with state-of-the-art methods. What's more, our method can be naturally extended to multi-view hand pose estimation by simply fusing depth data from multi views into a single point cloud.

Our main contributions are summarized as:

- We propose a novel end-to-end method named SHPR-Net to estimate 3D hand pose directly from point sets and demonstrate that it is highly effective and efficient.
- We propose a strategy that better fuses information from semantic segmentation network and regression network to achieve more representative features for hand pose estimation.
- We introduces a mechanism to deal with the challenges of geometric transformations by learning a transformation matrix for the input data and an inverse matrix for the output pose.

The remainder of this paper is organized as follows. Section II discusses related work. Section III introduces our proposed deep semantic hand pose regression network. Section IV provides experiments to compare our method with state-of-the-arts and discusses the impact of each module. Section V concludes this paper and points out some future work.

II. RELATED WORK

In this section we briefly review prior work that is highly related to this paper.

A. DEPTH-BASED HAND POSE ESTIMATION

Hand pose estimation can be generally categorized into three classes: model-based methods, discriminative methods and hybrid methods. Readers are referred to [22], [27], and [28] for more detailed review of hand pose estimation.

Model-based methods fit a predefined hand model into the input depth image to recover the model parameters of the input sample by optimizing an energy function. There are three important modules for model-based methods: optimization algorithm, hand model and energy function. Iterative closest point (ICP) [29] and particle swarm optimization (PSO) [30] are common choices for optimizing hand pose. Qian *et al.* [31] leveraged the properties of ICP and PSO to propose a new PSO-ICP method for minimizing energy functions. Several kinds of hand models were proposed [29], [31]–[35] to approximate the hand using spheres, cylinders and meshes etc. In most prior work, hand-crafted energy functions [29], [35]–[37] were employed to describe the difference between the input depth image and the current hand model. Despite of the good properties such as ensuring to output physically plausible poses, model-based methods need a strong prior (the predefined hand model) and complex optimization process, which is challenging for real-time applications.

On contrast, discriminative methods are usually totally data-driven. Some early work on discriminative hand pose estimation [36], [38]–[40] adopted random forest to predict the hand pose. As the successful applications on many different fields of deep learning, the majority of recent work of hand pose estimation has shifted to deep learning, especially CNN based methods. One family of solutions uses CNNs to predict heatmap of each joints and employs post-processing to recover the hand pose [41], [42]. Another line of work focuses on directly predicting the 3D coordinates of hand joints via regression [7], [8], [11], [14], [43]. Oberweger and Lepetit [11] and Oberweger *et al.* [43] leveraged the fact that hand pose actually lies in a low dimension manifold and proposed DeepPrior/DeepPrior++ to first predict the hand parameters in low dimension and project back to original hand pose domain using principle component analysis (PCA). Guo *et al.* [7] and Wang *et al.* [14] proposed a region ensemble network (REN) by dividing feature maps into several regions and fusing regional features to predict the final hand pose. Chen *et al.* [8] exploited a cascaded framework upon REN to iteratively mining more discriminative features under the guidance of previously predicted hand pose. 3D CNNs were also explored for hand pose estimation [9], [19], [23] to better leverage the spatial information of depth data. Moon *et al.* [19] proposed the V2V-PoseNet to exploit voxel-to-voxel predictions for hand pose from 3D volumetric forms and adopted epoch ensemble strategy that averages the predictions of ten models.

Hybrid methods try to take advantage of the properties of model-based and discriminative methods. Ye *et al.* [12] proposed a spatial attention network to produce initial predictions and exploited hierarchical partial PSO to enforce kinematic constraints for the predicted hand pose. Zhou *et al.* [44] proposed to incorporate an explicit hand model into CNNs to ensure predicting physically plausible hand poses. Malik *et al.* [45] extended the deep hand model method to simultaneously estimate bone-lengths of hand skeletons and hand poses. Dibra *et al.* [46] proposed to use CNNs and hand mesh model to refine 3D hand pose from unlabeled data on top of the model pre-trained on synthetic depth images.

Our method falls into the category of discriminative methods. Different from prior work that utilizes 2D depth images or 3D volumetric representations, our proposed SHPR-Net directly processes on point sets to predict hand poses.

B. DEEP LEARNING ON POINT SETS

Deep learning directly on point sets is a challenging problem since point sets suffer from the properties of unordered input, geometric transformation etc. PointNet [25] is a pioneer in this field. PointNet learns point-wise features using multi-layer perceptron and employs a symmetric function to obtain global features that are invariant to input permutations. Though effective, by design PointNet does not fully leverage the local information between points. PointNet++ [26] is an extended version of PointNet to hierarchically capture local structure patterns on a nested grouped partitioning of input point set. Recently, more architectures are proposed to improve feature learning on point sets in different aspects, such as PointCNN [47], Pointwise CNN [48], DGCNN [49], SO-NET [50] etc. Though these work shows impressive performance on 3D object classification, object part segmentation and large scene semantic segmentation tasks, there are few attempts to focus on 3D articulated hand pose estimation from point sets except [21]. However, Hand PointNet [21] is not totally end-to-end since it consists of the pre-processing to normalize point clouds and a post-processing network to refine fingertips. Our SHPR-Net is an end-to-end network and differs from Hand PointNet [21] in several aspects. We address the problem of geometric transformation by incorporating transformation matrices in input and output spaces and train the whole network in end-to-end manner. What's more, we leverage the semantic information to better extract features from point cloud for hand pose regression.

C. GEOMETRIC TRANSFORMATION

Geometric transformation is a common challenge for computer vision problems since it increases the diversity of input data. Spatial Transformer Network (STN) [51] is one of the most important researches to tackle this problem. It explicitly incorporates a spatial transformer module to empower CNNs to be invariant to generic transformations of input data. Shi *et al.* [52] proposed a Progressive Calibration Network (PCN) to detect rotation-invariant faces in multiple stages with coarse-to-fine strategy. Ye *et al.* [12] also adopted the spatial attention mechanism to the problem of hand pose estimation to reduce the viewpoint and articulation variations. However, this mechanism is applied on 2D space and only considers in-plane rotations. PointNet [25] proposed to use a mini-network to predict an affine transformation matrix and apply it to the input point cloud or intermediate features. This method was proved to be quite effective for object classification and semantic segmentation tasks. However, unlike these tasks, hand pose regression does not desire invariance to the transformations of input data thus directly applying transformation matrix in input or feature space would not help. Ge *et al.* [21] normalized the rotations of point sets using an oriented bounding box by performing principle component analysis (PCA). However, it's not always easy to determine the orientation of point clouds by PCA, especially when the point clouds undergo incomplete points and noises.

To address this problem, in this work we propose to learn two transformation matrices and apply them to the input and output points respectively. To make sure the transformation matrix of the output space is an inverse matrix of the input one, we propose the identity matrix loss to enforce the multiplication of these two matrices to be close to identity matrix. In this manner the network learns to transform original point clouds into latent canonical spaces. The transformation matrix is learned together with the hand pose in an end-to-end manner, which may benefit the robustness of finding an optimal transformation.

D. COUPLING SEMANTIC INFORMATION WITH REGRESSION

Early work on pose estimation [38], [53] first predicts part segmentation label for each pixel and recovers pose coordinations from these semantic labels. Another family of algorithms [7], [39], [43], [54] formulated pose estimation as a regression problem without intermediate semantic segmentation. Some researches have explored the combination of semantic segmentation and holistic regression. Neverova *et al.* [55] proposed a semi/weakly-supervised way to learn an intermediate representation in the form of segmentation map. Then the segmentation information is used to extract local regions from features map to enhance the original features in the regression network. Wan *et al.* [18] performed 2D heatmap detection, 3D heatmap detection and dense 3D unit vector regression in a multi-task setup. In this paper, we propose a new method to incorporate semantic information into the regression network, which fuses the semantic labels and raw point sets in the input and also fuses semantic information and latent features in the output of the regression network.

III. SHPR-Net: DEEP SEMANTIC HAND POSE REGRESSION NETWORK

Our work deals with the problem of hand pose estimation directly from point clouds. The framework of our proposed method (SHPR-Net) is sketched in Fig. 1. SHPR-Net takes

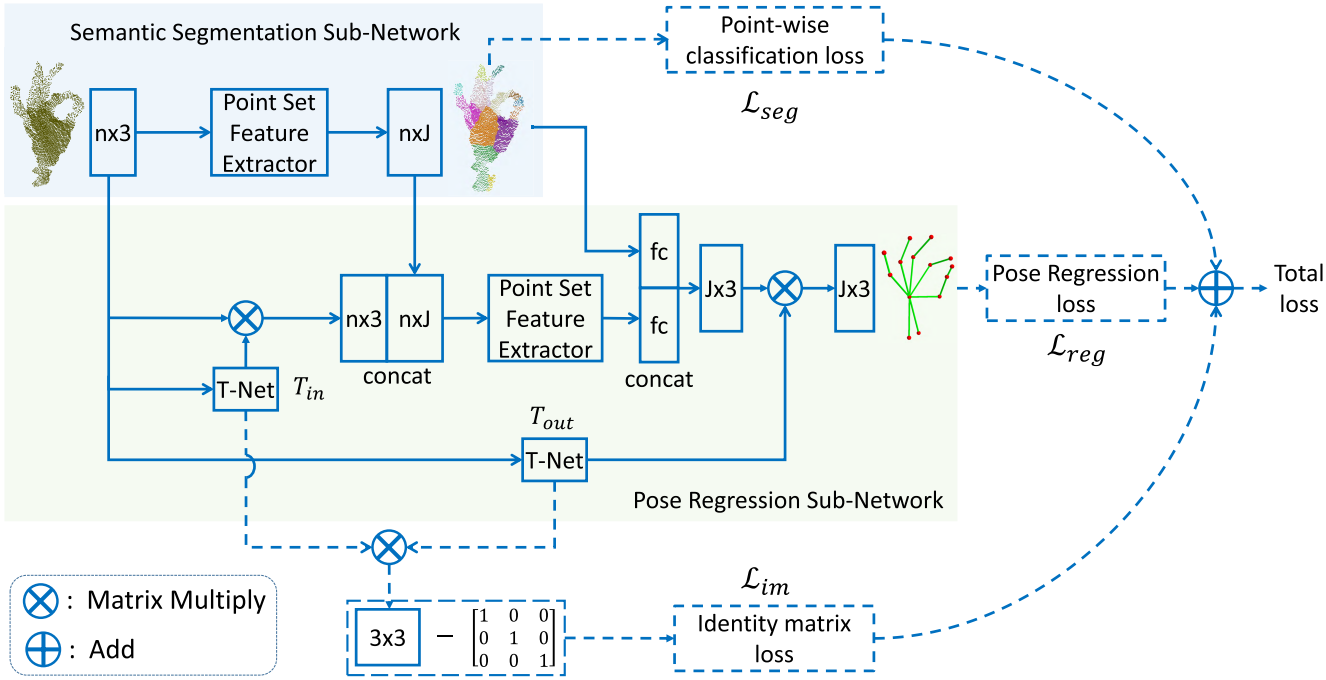


FIGURE 1. The framework of the proposed method (SHPR-Net). All dotted diagrams are only used in training phase and others are included in testing phase. Our method consists of two subnetworks: the semantic segmentation network (SegNet) and the hand pose regression network (RegNet). SegNet produces semantic labels for each point with the help of a point set feature extractor. The semantic labels are then fused in the input and output layers of the RegNet. Two mini PointNets (T-Nets) predict two transformation matrices to transform input points into latent canonical space and transform output pose back to original space. The identity matrix loss, together with point-wise classification loss and pose regression loss are utilized to train the whole network in an end-to-end manner. We adopt PointNet++ [26] as point set feature extractor in this paper.

a point cloud as input and predicts the corresponding hand pose. The input point cloud goes through a semantic segmentation network (SegNet) to obtain semantic labels. Two transformation matrices T_{in} and T_{out} are learned from the input point cloud. The original point cloud is transformed by T_{in} and concatenated with semantic labels before being fed into the hand pose regression network (RegNet). The semantic labels are also fused in the fully connected layers of RegNet. The hand pose is predicted by RegNet and transformed by T_{out} to obtain the final prediction. The whole SHPR-Net is trained in an end-to-end manner to minimize the addition of point-wise classification loss, pose regression loss and identity matrix loss.

As shown in Fig. 1, a point set feature extractor is adopted to process the input point set and produce features. Generally, every algorithm that works on point set can be chosen here but a complete exploration of more choices is beyond the focus of this paper. In this paper, we employ PointNet++ [26] as the backbone architecture to extract features from point sets. In this section we will first give a brief review of PointNet/PointNet++ and provide elaborations of our proposed method.

A. PRELIMINARY

Suppose $\{x_i\}_{i=1}^N$ is the input point set, where N is the number of points. The goal of PointNet [25] is to extract global feature vector f_g that are invariant to input permutation, which is achieved by a symmetric function $\mathcal{F}$.

$$f_g = \mathcal{F}(h(\{x_i\}_{i=1}^N)), \tag{1}$$

where h is a multi-layer perceptron (MLP) network to learn point-wise features.

Applying fully connected (fc) layers on top of global features leads to object classification:

$$c = fc(f_g). \tag{2}$$

For object part segmentation and scene semantic segmentation, point-wise local features are concatenated with the global features to produce semantic labels.

PointNet++ [26] is an extension of PointNet to hierarchically extract features from point sets, like the structure of CNNs. Briefly, in each layer PointNet++ samples and groups points into several subsets and applies PointNet on each group of points to extract features. Stacking several layers leads to the architecture of PointNet++.

Readers are referred to [25] and [26] for more details of PointNet and PonitNet++.

B. SEMANTIC SEGMENTATION NETWORK (SegNet)

The semantic segmentation network (SegNet) is used to predict semantic label for each input point. Formally, given a point set $\{x_i\}_{i=1}^N$, the output of SegNet is $\{p_i\}_{i=1}^N$, where $p_i \in \mathcal{R}^{1\times J}$ is the probability of ith point belonging to different

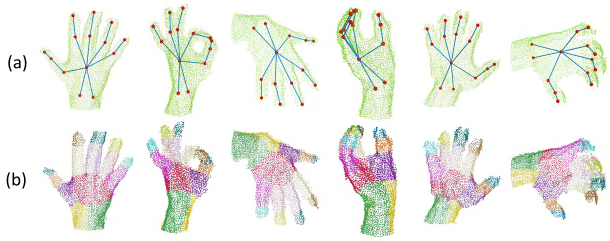


FIGURE 2. Samples of semantic segmentation for hand point clouds. (a) Point clouds with ground truth hand pose. (b) Inferred semantic labels from annotated hand pose, as in (3).

categories and J is the number of joints in hand skeleton. Applying a softmax function on $\{p_i\}_{i=1}^N$ leads to the final semantic labels $\{c_i\}_{i=1}^N$, where $1 \leq c_i \leq J$, and c_i is the predicted semantic label for i^{th} point.

We follow the network architecture with single scale grouping of part segmentation task in [26] to design SegNet. SegNet consists of three abstraction levels which extract features for local regions. Three feature propagation layers are exploited to concatenate features from different levels to obtain point features for all original points. The detailed architecture of SegNet can be found in Fig. 11 in Appendix.

For hand pose estimation task, the coordinates of hand joints are given for each sample and the per-point semantic labels are usually not provided. We derive semantic labels from the annotated hand pose for training (see Fig. 2). Specifically, given the annotated hand pose $\{\tilde{y}_k\}_{k=1}^J$, the ground truth of part label $\tilde{c}_i$ for point x_i can be obtained as:

$$\tilde{c}_i = \arg\min_{k \in [1,J]} ||x_i - \tilde{y}_k||^2. \tag{3}$$

We then calculate cross entropy loss (denoted as $\mathcal{L}_{seg}$) for each p_i and c_i^{gt} as the objective function of point-wise semantic segmentation network:

$$\mathcal{L}_{seg} = -\sum_{k=1}^{J} \mathbb{1}_{\tilde{c}_i = c_i} log(p_{i,k}), \tag{4}$$

where $\mathbb{1}_{\tilde{c}_i = c_i}$ is an indicator function that equals to one if $\tilde{c}_i = c_i$ and zero otherwise, $p_{i,k}$ is the probability of ith point belonging to k^{th} category.

C. HAND POSE REGRESSION NETWORK (RegNet)

The hand pose regression network (RegNet) aims to directly estimate the 3D coordinates of hand pose from the input point set. A basic regression network can be obtained by replacing the output layer of the classification PointNet/PointNet++ by a fully connected layer that predicts the coordinates of hand pose. Our proposed RegNet goes further in two aspects: geometric transformation and semantic enhancement.

The original PointNet [25] introduced a mini-network (T-Net) to estimate an affine transformation matrix to deal with the problem of point cloud transformation. T-Net resembles the architecture of PointNet but is relatively smaller. By incorporating T-Net into input and feature spaces, PointNet can achieve invariance to geometric transformation of input point set and significantly improves the performance. However, unlike classification and segmentation tasks, the output of hand pose regression will change accordingly when the input point set undergoes certain geometric transformations. Therefore, directly learning to transform the input points and features is not an appropriate way for regression network. To address this problem, we incorporate two T-Nets to learn two matrices (T_{in} and T_{out}) to transform the input points and output pose respectively. Ideally, T_{out} should be the inverse matrix of T_{in} to maintain the geometric properties. To this end, we constrain the multiplication of T_{in} and T_{out} to be close as an identity matrix by introducing the identity matrix loss:

$$\mathcal{L}_{im} = ||T_{in}T_{out} - I||^2, \tag{5}$$

where I is an identity matrix. This will not bring heavy computational and memory burdens to RegNet because $T_{in} \in R^{3\times3}$ and $T_{out} \in R^{3\times3}$ are low dimensional matrices.

PointNet [25] lacks of capability to capture local information among different points. PointNet++ [26] relieves this issue by hierarchically extracting local features of overlapping groups. However, this strategy still has limited ability to perceive the information of hand poses as the relations of different joints are not considered in the grouping operation. Therefore, we propose a new method to incorporate semantic information into hand pose regression network to solve the above problems. Firstly, we concatenate the coordinates of point set with the semantic probabilities produced by SegNet and feed it to the RegNet. To better fuse the information of SegNet, the predicted semantic probabilities are passed through a fully connected (fc) layer and then concatenated with the output of the first fc layer of RegNet. The fused features undergo another two fc layers to produce the predicted hand pose.

RegNet has similar backbone architecture as the classification network with single scale grouping of [26], the detailed architecture of RegNet can be found in Fig. 12 in Appendix.

Similar to previous regression based methods [7], [8], we use smooth L1 loss [56] $\mathcal{L}_{smoothl1}$ between the predicted hand pose $y = \{y_k\}_{k=1}^J$ and the ground truth pose $\tilde{y} = \{\tilde{y}_k\}_{k=1}^J$ for regression network:

$$\mathcal{L}_{reg} = \mathcal{L}_{smoothl1}(y, \tilde{y}). \tag{6}$$

D. TRAINING LOSS

We train the whole network in an end-to-end manner by minimizing the total training loss:

$$\mathcal{L}_{total} = \mathcal{L}_{reg} + \alpha\mathcal{L}_{im} + \beta\mathcal{L}_{seg}, \tag{7}$$

where α and β are weighted terms of identity matrix loss and semantic segmentation loss respectively.

In this equation, $\mathcal{L}_{seg}$ penalizes the errors of point-wise classification, $\mathcal{L}_{reg}$ penalizes the errors of output hand pose while $\mathcal{L}_{im}$ enforces the inverse property of input and output transformation matrices.

TABLE 1. Comparison of the proposed method with state-of-the-art methods on ICVL [39], MSRA [40] and NYU [41] dataset. *It should be noted that in table (c), the entry SHPR-NET(Ours, three views) uses depth data from three views while all other methods are conducted on single (frontal) view of NYU dataset, see Section IV-A3 for detailed discussions. (a) ICVL. (b) MSRA. (c) NYU.

Methods	Error (mm)
LRF [39]	12.56
DeepModel [44]	11.56
DeepPrior [43]	10.4
CrossingNets [10]	10.2
JTSC [57]	9.16
DeepPrior++ [11]	8.1
REN (4x6x6) [7]	7.63
REN (9x6x6) [14]	7.31
DenseReg [18]	7.3
Hand PointNet [21]	6.9
Pose-REN [8]	6.79
V2V-PoseNet [19]	6.28
SHPR-Net (Ours)	7.22

(a)

Methods	Error (mm)
Cascaded [40]	15.2
Multi-view CNNs [42]	13.1
Madadi *et al.* [58]	12.8
Baek *et al.* [20]	12.5
CrossingNets [10]	12.2
REN (9x6x6) [14]	9.79
3DCNN [9]	9.58
DeepPrior++ [11]	9.5
Pose-REN [8]	8.65
Hand PointNet [21]	8.5
V2V-PoseNet [19]	7.59
DenseReg [18]	7.23
SHPR-Net (Ours)	7.76

(b)

Methods	Error (mm)
DISCO [59]	20.7
DeepPrior [43]	19.73
DeepModel [44]	16.90
JTSC [57]	16.80
Feedback [60]	15.97
Madadi *et al.* [61]	15.60
CrossingNets [10]	15.5
Neverova *et al.* [55]	14.94
Lie-X [62]	14.51
Baek *et al.* [20]	14.1
3D CNN [9]	14.11
REN (4x6x6) [7]	13.39
REN (9x6x6) [14]	12.69
DeepPrior++ [11]	12.24
Pose-REN [8]	11.81
Hand PointNet [21]	10.5
DenseReg [18]	10.21
V2V-PoseNet [19]	8.42
SHPR-Net (Ours, frontal view)	10.78
SHPR-Net (Ours, three views)*	9.37

(c)

E. MULTI-VIEW DEPTH-BASED HAND POSE ESTIMATION

Hand pose estimation from multi-view depth images has rarely been explored in literature. Due to the representation of input data, our method can be naturally extended to handle the multi-view scenarios. This can be achieved by simply fusing depth data from different views into a single point cloud and feeding the fused points into our SHPR-Net. No additional modifications are required except for the input data fusion. We will demonstrate the capability of our method to handle multi-view data by conducting experiments on NYU dataset in Section IV-A3.

F. IMPLEMENTATION DETAILS

The proposed method is implemented in Tensorflow [63]. Experiments are conducted on a server with two Intel Xeon E-2640 CPUs, 256GB RAM and four NVIDIA Geforce 1080TI GPUs.

1) PREPROCESSING

We first convert all pixels in depth image to world coordinates to generate a point cloud. We follow similar strategy as prior work [7], [8], [11] to crop the hand region with a 3D cube with the size of $240mm^3$. Since the cropped point cloud has different number of points due to different distances to the camera, we use Poisson Disk Sampling algorithm [64] provided by Meshlab [65] to sample the original point cloud to approximately certain number of points. The coordinates of sampled points and the corresponding annotated hand poses are then normalized into $[-1, 1]$.

2) PARAMETER SETTINGS

We use $N = 4096$ points in the experiments. The impacts of different number of points will be discussed in Section IV-B3. We set $\alpha = 0.001$ and $\beta = 0.001$ for all experiments.

3) DATA AUGMENTATION

We apply online random data augmentation strategy to increase generalization ability of the network. Specifically, we apply random rotation along z axis with the range of $[-15°, 15°]$, random scaling of $[0.9, 1.1]$ and random translation of $[-0.005mm, 0.005mm]$ to the point sets and corresponding annotated hand poses in training.

4) TRAINING

We use Adam [66] optimizer to train the network using four GPUs, with the total batch size of 32. The initial learning rate is set to 0.001 and multiplied with 0.7 after each 10 epochs. The whole network was trained for 100 epochs.

IV. EXPERIMENTS

We evaluate our proposed method on three public hand pose datasets: ICVL dataset [39], NYU dataset [41] and MSRA dataset [40].

ICVL dataset was captured with an Intel Realsense Camera and contains 330k training samples from 10 different subjects. There are 1596 samples in the test set. Each depth image is annotated with a hand pose with $J = 16$ joints, including 1 palm joint and 3 joints for each fingers.

NYU dataset was captured with three Microsoft Kinects that placed at different views. It consists of 72k samples for training and 8252 samples for testing. There are two subjects in the test set while only one appears in the training set. The annotated hand pose has 36 joints and we follow the evaluation protocol of prior work to use only $J = 14$ of them in the experiments.

MSRA dataset contains 76k frames from 9 different subjects with 17 different gestures. The leave-one-subject-out cross-validation strategy is employed for evaluation, similar

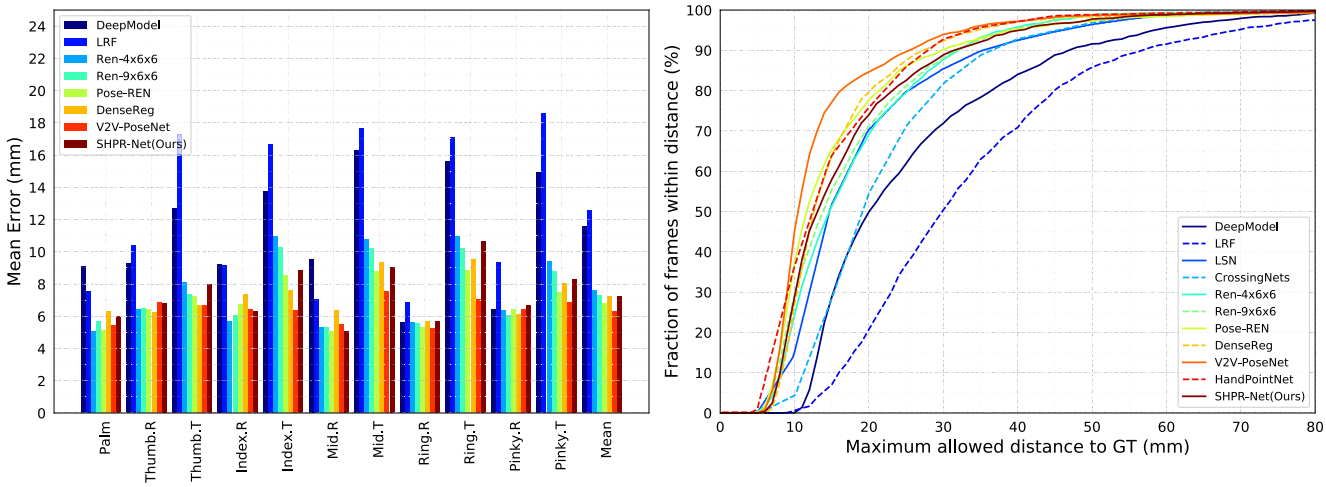


FIGURE 3. Comparison with state-of-the-arts on ICVL [39] dataset. Left: per-joint errors. Right: the proportion of good frames over different error thresholds.

as all prior work. There are $J = 21$ joints in each annotated hand pose, with 4 joints in each finger and 1 palm joint.

Following the most commonly used metrics in literature, we evaluate the proposed method with two different metrics. First, we report the average 3D joint error of each joint as well as average error of total joints over all testing frames. This metric indicates the overall performance of hand pose estimation and also show performances for different joints. The second metric is the fraction of frames whose errors of all joints are within a threshold. This is a more challenging and strict metric that better presents the performance of a hand pose estimator.

In the following parts of this section, we first compare our proposed method with state-of-the-art methods. After that we provide extensive self comparison experiments to demonstrate the impacts of different modules or design choices of our method. Finally some visualization results bring more insights and better understanding.

A. COMPARISON WITH STATE-OF-THE-ARTS

We compare our proposed method with tens of prior methods, including latent random forest (LRF) [39], DISCO nets [59], cascaded hand pose regression (Cascaded) [40], 2D CNN with deep hand model (DeepModel) [44], 2D CNN with priors (DeepPrior) [43] and its improved version (DeepPrior++) [11], 2D CNN with feedback loop (Feedback) [60], Lie group based method (Lie-X) [62], Neverova *et al.* [55], multi-view 2D CNNs (Multiview) [9], joint training with shared context methods (JTSC) [57], variants of region ensemble network (REN-4x6x6 [7] and REN-9x6x6 [14]), pose guided structured REN (Pose-REN) [8], CrossingNets [10], dense 3D regression (DenseReg) [18], 3D CNN [9], 3D CNN with voxel-to-voxel predictions (V2V-PoseNet) [19], hierarchical PointNets based hand regression (Hand PointNet) [21], and Baek *et al.* [20].

1) ICVL DATASET

On ICVL dataset, we compare our proposed method with [7], [8], [10], [11], [14], [18], [19], [21], [39], and [44]. The average errors for different joints and the proportion of good frames over different error thresholds are shown in Fig. 3. We also report the average error over all joints and compare with prior methods, as shown in Table 2a. These results indicate that our method outperforms most of state-of-the-art methods and is on par with the rest of them. As demonstrated in prior work [18], ICVL dataset has considerably achieved nearly saturated average joint error, which makes the gaps between our method and [8], [19], and [21] seem less significant.

2) MSRA DATASET

On MSRA dataset, we compare our method against several prior methods [8], [9], [11], [14], [18], [19], [21], [40]. The fraction of success frames with respect to maximum allowed threshold and per-joint errors are shown in Fig. 4. The comparison of average error over all joints are given in Table 2b. As can be seen in Fig. 4, our method outperforms [8], [9], [11], [14], [40] and achieves quite comparable performance with DenseReg [18]. Our method performs better than V2V-PoseNet [19] when the error threshold is larger than 18 mm. Table 2b provides clearer comparisons. Our method obtains average joint error that is about 5.1 mm, 4.2 mm, 1.8 mm, 1.6 mm, 0.7 mm, 0.5 mm smaller than [8]–[11], [21], [14], and [42] respectively and get comparable performance with [19] and [18].

Following the evaluation protocol of prior work [8], [9], [40], we also plot the mean joint error over different viewpoint angles, as shown in Fig. 5. As can be seen, our method performs better than [8], [9], and [14] in almost all viewpoints and can obtain considerably good results in large yaw and pitch angles, which demonstrates the strong performance and robustness of our proposed method.

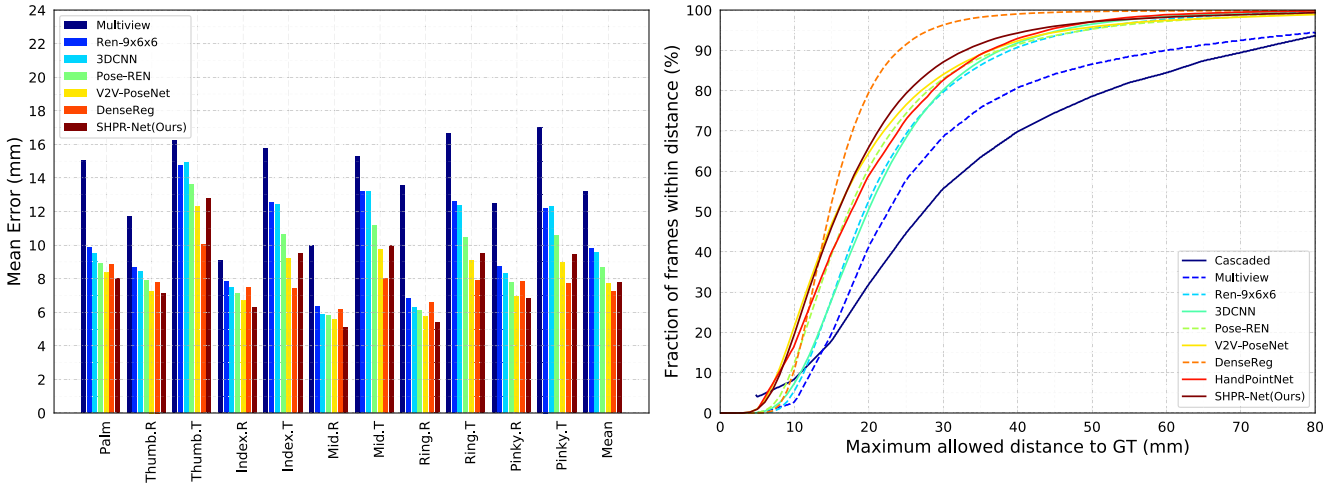


FIGURE 4. Comparison with state-of-the-arts on MSRA [40] dataset. Left: per-joint errors. Right: the proportion of good frames over different error thresholds.

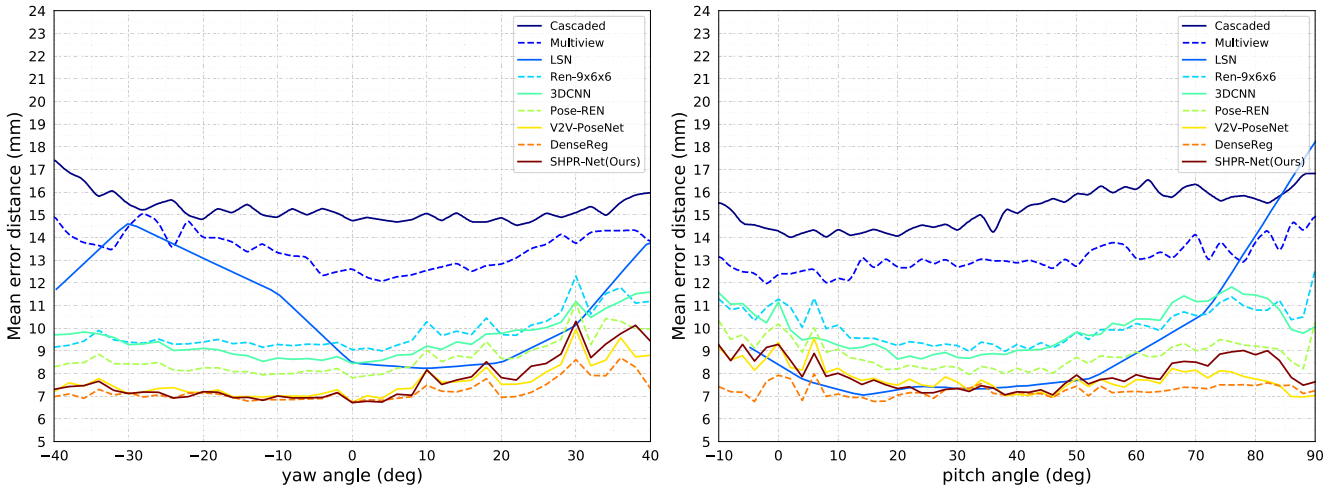


FIGURE 5. Comparison of mean error distance over different yaw (left) and pitch (right) viewpoint angles on MSRA [40] dataset.

3) NYU DATASET

NYU dataset is a relatively more challenging benchmark as it exhibits more complex articulations, sensor noises and hand shape variations etc. We compare our method on NYU dataset with [7]–[11], [14], [18], [19], [21], [43], [44], [55], [57], and [59]–[62].

As shown in Fig. 6, with point clouds from a single view (frontal view), our method shows comparable performance with DenseReg [18] and Hand PointNet [21], slightly worse performance than V2V-PoseNet [19] and outperforms the rest of state-of-the-arts.

We also compare the average error of all joints with state-of-the-arts in Table 2c. As can be seen, our method achieves top performance among state-of-the-arts, which demonstrates the effectiveness of our method.

Multi-View Experiments on NYU Dataset: NYU dataset provides depth images from three cameras that are placed at different viewpoints (two side views and one frontal view). However, the calibration parameters of three cameras are not provided and the cameras are moved occasionally in data capturing. We use the annotated hand poses from different views to calibrate the cameras. Specifically, we calculate the transformation matrices that project annotated hand joints from two side views to frontal view. We then use these matrices to project point clouds of side views to frontal view and fuse all points from three views to generate a single point cloud. It's worth noting that in real scenarios, the calibration parameters between cameras can be easily obtained by an offline calibration procedure and do not require any hand pose annotations.

Some examples of fused point clouds are shown in Fig. 7. Each pair of point clouds presents the points from frontal view (left) and three views (right) respectively. It can be observed that by fusing depth data from three views, the point clouds are more complete and robust to occlusions. However, the fused point clouds also suffer from slightly heavier noises,

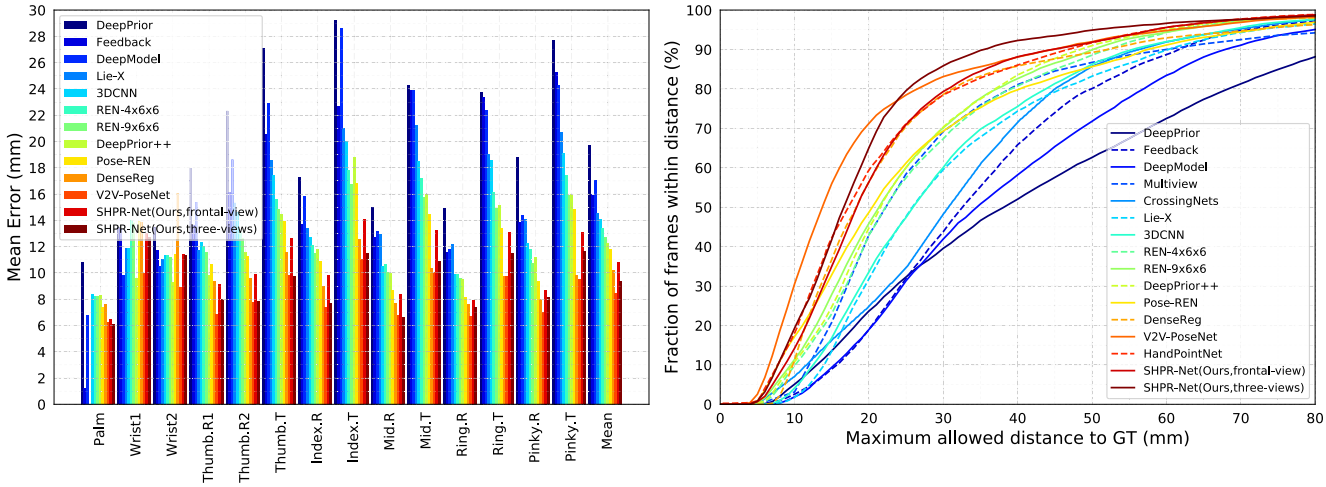


FIGURE 6. Comparison with state-of-the-arts on NYU [41] dataset. Left: the proportion of good frames over different error thresholds. Right: per-joint errors.

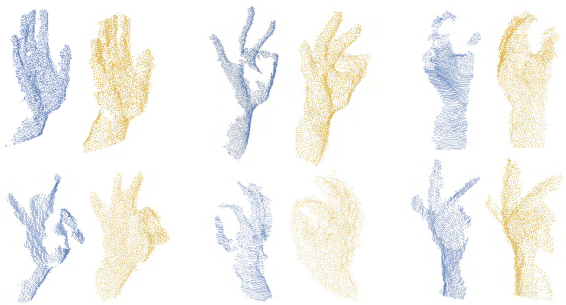


FIGURE 7. Examples of fused point clouds from three views of NYU [41] dataset. For each pair of point sets, the left one is from the frontal view and the right one shows the fused point cloud from three views.

which is probably due to the imperfect calibrations between cameras and noises in depth images.

As shown in Table 2c, when using depth data from three views, our method achieves the second best performance among all methods when regards to average joint error. Fig. 6 further shows the fraction of frames whose maximum joint error falls within a threshold, which is a more challenging metric. With multi-view depth data, our method outperforms all state-of-the-arts when the error threshold is bigger than about 23 mm. For example, the proportion of good frames of our method is about 5% more than DenseReg [18] and V2V-PoseNet [19] when the error threshold is 40 mm. When the error threshold is smaller than 23 mm, our method performs slightly worse than V2V-PoseNet [19] but still outperforms others.

It's worth noting that this is not direct comparison because none of existing methods have explored the usage of multi-view data. However, in this experiment we demonstrate that our method can achieve further improvement by exploiting multi-view depth data without any modifications to the network. We expect that this observation would be beneficial to the community of hand pose estimation.

B. ABLATION STUDY

In this section, we will provide extensive experiments to analyze the impacts of different modules of SHPR-Net and different design choices. Unless otherwise stated, all ablation studies are conducted on multi-view depth data of NYU dataset.

1) IMPACTS OF INPUT AND OUTPUT TRANSFORMATIONS

To evaluate the impacts of our proposed input and output transformation, we conduct experiments with different variants of regression PointNet by inserting different transformation modules. Specifically, we explore different combinations of input transformation, feature transformation, output transformation and identity matrix loss.

As shown in Table 2, simply inserting input transformation into PointNet improve little due to the inconsistence of geometric transformation between input point cloud and output pose. The improvement is probably due to more model parameters brought by the T-Net. The performance degrades when incorporating additional feature transformation because it further increases the inconsistence of transformation. When we insert input and output transformations into PointNet but do not explicitly constrain the relations between these two matrices, the performance is improved due to the inherent potential to learn consistent transformation. However, the improvement is still insignificant because the constraints between input and output transformation are

TABLE 2. Impacts of input and output transformation. Self comparisons on NYU [41] dataset with depth data from three views.

Input	Feature	Output	Identity matrix loss	Mean error (mm)
				15.07
✓				14.23
✓	✓			15.70
✓		✓		13.79
✓		✓	✓	**11.39**

not explicitly given. Further applying identity matrix loss to the network significantly boost the performance, as the output transformation matrix tends to be the inverse matrix of input transformation matrix, which well preserve the nature of regression task. The above observations demonstrate that learning constrained transformation matrices for input and output space is an effective way to handle the geometric transformation for hand pose regression problem.

TABLE 3. Impacts of segmentation task. We report average joint errors (mm) on NYU [41] dataset with depth data from three views.

Methods \ Backbone	PointNet	PointNet++
w/o SegNet	11.39	10.31
w/ SegNet	10.69	**9.37**

2) IMPACTS OF SEMANTIC INFORMATION

We evaluate the impacts of semantic information by comparing the performance of our method with or without the SegNet. As shown in Table 3, when we use PointNet as the backbone architecture to extract features for point sets, coupling semantic segmentation task with regression task reduces the error of hand pose regression by 0.7 mm. When we switch to PointNet++ as the backbone architecture, the gap brought by semantic information becomes 0.94 mm, which is probably due to the more powerful architecture of PointNet++. These experiments demonstrate the effectiveness of our proposed strategy to incorporate semantic information into regression network.

3) IMPACTS OF POINT NUMBER

To evaluate how the number of points N in a point set affects the performance of the proposed method, we conduct experiments with different numbers of points in the input point cloud. As shown in Table 4, When we use only 512 points, the performance of SHPR-Net is still quite competitive. The average joint error drops by 0.47 mm, 0.09 mm when N increases from 512 to 1024 and 1024 to 2048 respectively. Further increasing N to 4096 still brings performance improvement. To balance accuracy and computational complexity, we choose $N = 4096$ and do not exploit more points.

TABLE 4. Impacts of point number. Self comparisons on NYU [41] dataset with depth data from three views.

Point number	Mean error (mm)
512	10.05
1024	9.58
2048	9.49
4096	**9.37**

TABLE 5. Impacts of multi-view point cloud fusion. Self comparisons on NYU [41] dataset.

Views \ Backbone	PointNet	PointNet++
frontal view	12.02	10.78
frontal + 2 side views	10.69	**9.37**

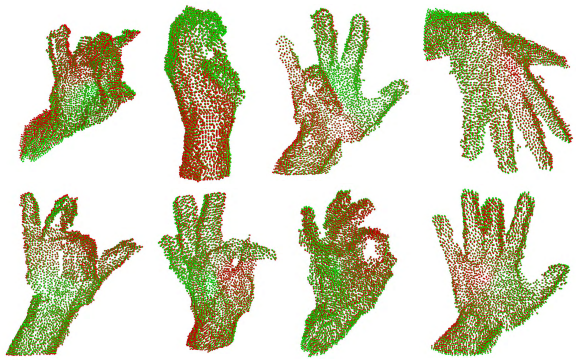


FIGURE 8. Visualization of the effect of identity matrix loss. We visualize the original point clouds in red and the transformed point clouds which undergo input and output transformation matrices in green. It can be seen that the transformed point clouds almost completely overlap with the original ones, which illustrates the effectiveness of identity matrix loss.

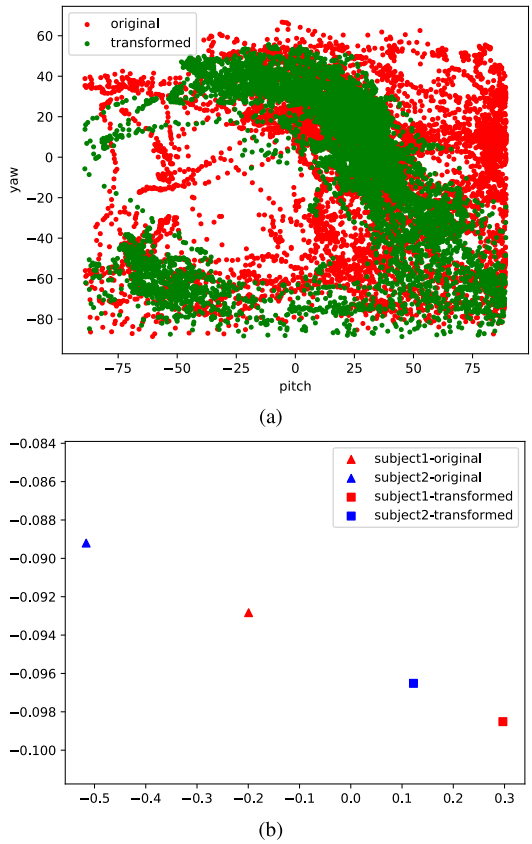


FIGURE 9. Visualization of the impacts of input transformation matrix on NYU [41] dataset. (a) The viewpoint distributions of original point clouds and the transformed point clouds is visualized in red and green respectively. The viewpoints of transformed point clouds distribute more compact than the original ones. (b) We visualize the average hand shapes for two subjects of original and transformed point clouds. It can be seen that after applying input transformation, the distance between different subjects is reduced, which makes our method more robust to hand shape variations.

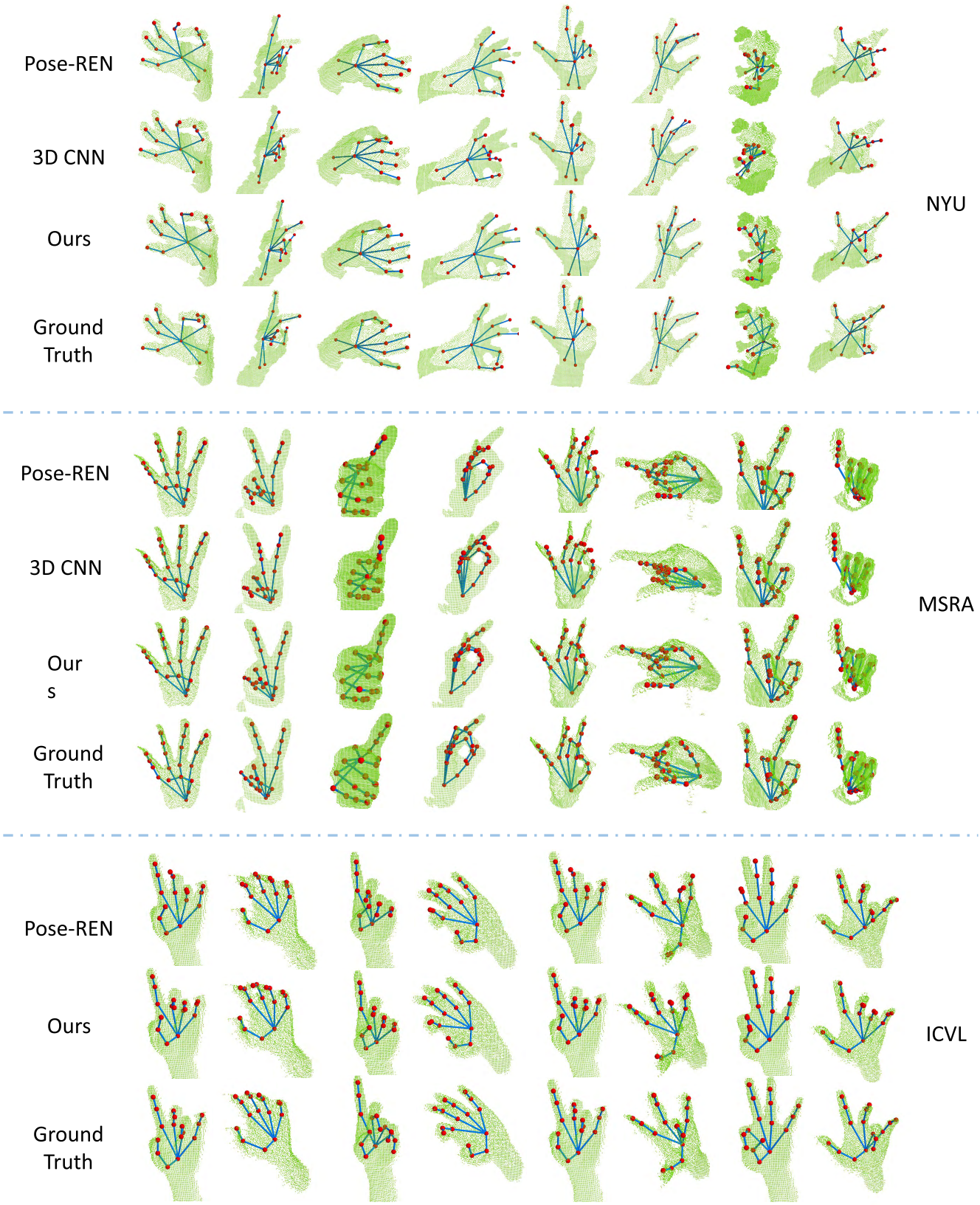


FIGURE 10. Qualitative results on NYU, ICVL and MSRA dataset. For each dataset, we visualize and compare the predictions of Pose-REN [8], 3D CNN [9] and our proposed SHPR-Net.

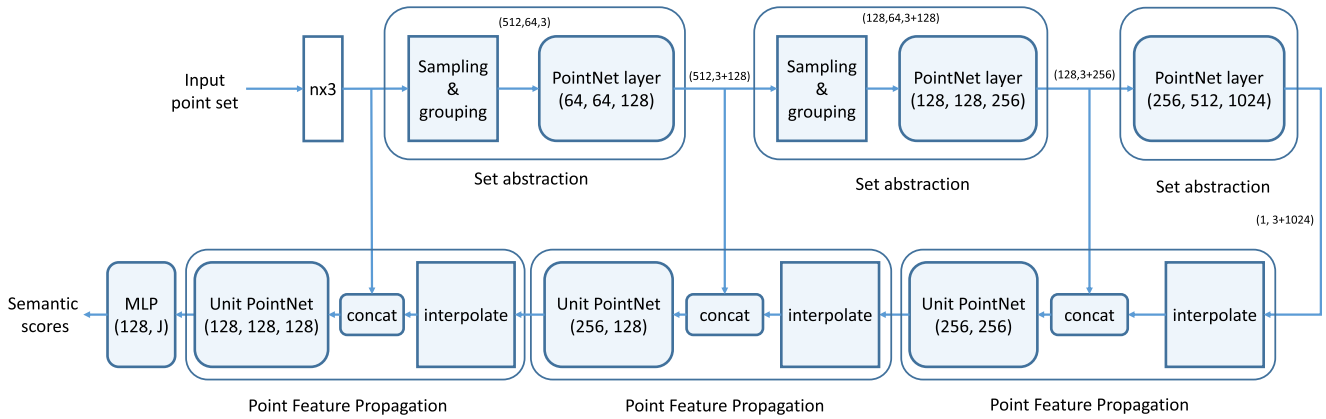


FIGURE 11. The detailed architecture of SegNet.

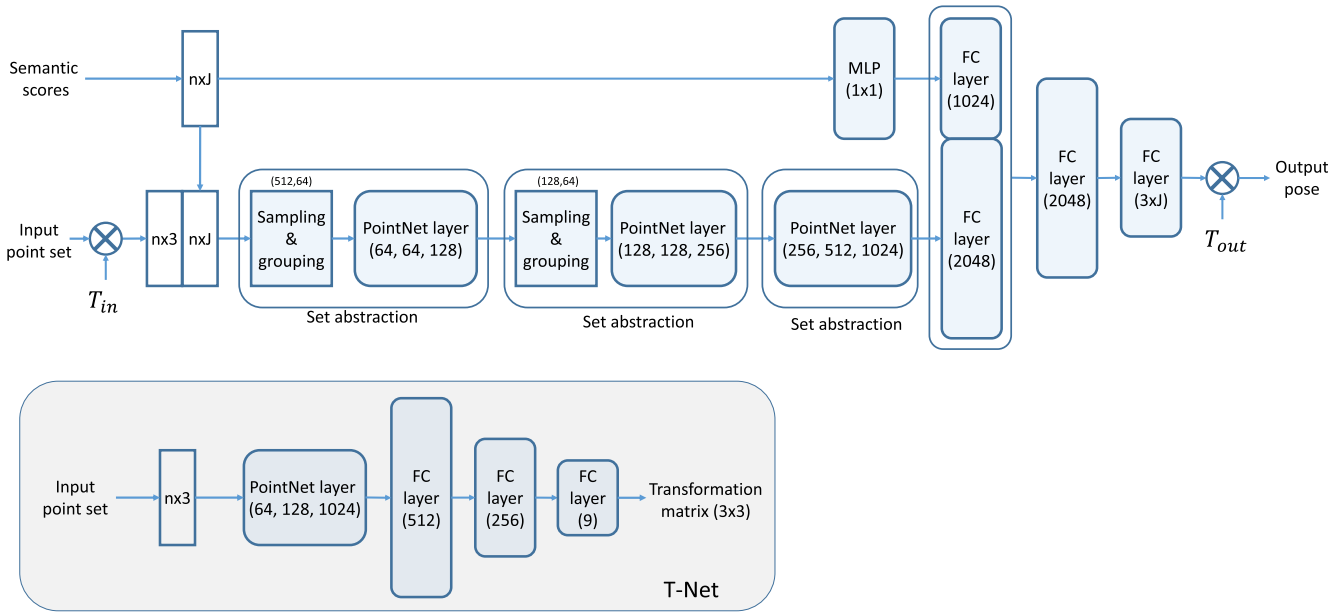


FIGURE 12. The detailed architecture of RegNet.

4) IMPACTS OF MULTI-VIEW POINT CLOUD FUSION

To further explore the impacts of using point clouds from multi-views, we conduct experiments with different views of depth data on top of our method. As shown in Table 5, when using point clouds from three views, the performance improves by about 1.33 mm and 1.41 mm for PointNet and PointNet++ backbones respectively. This indicates the potentials to exploit multi-view data to boost the accuracy of hand pose estimation. Our method can naturally extend to this scenario without any modifications to the network.

C. VISUALIZATIONS

1) VISUALIZING THE IMPACTS OF IDENTITY MATRIX LOSS

To demonstrate how the constraint of identity matrix loss works, we apply the input and output transformation matrices on point clouds and visualize the transformed and original ones. We expect that the output transformation matrix is the inverse matrix of the input one. Therefore, undergoing the input and output matrices should produce a point cloud that is the same with the original one. As shown in Fig. 8, the transformed point sets (green) almost totally overlap with the original ones (red), which indicates that identity matrix loss is an effective way to produce a transformation matrix and an correspondingly inverse transformation matrix.

2) VISUALIZATION OF INPUT TRANSFORMATION

Ideally the input transformation will transform the input point set into a latent canonical space in which the variations of viewpoints and hand shapes are greatly reduced, so that the network can be easier to learn a good regressor for hand pose.

We visualize the distributions of viewpoints before/after the input transformation in Fig. 9a. In this figure we can observe that the viewpoints of transformed point clouds (green) distribute more compact than the original ones (red), which indicates the effectiveness of the strategy of applying geometric transformation on input point sets. What's more, we visualize the average hand shapes for two subjects in NYU dataset of original and transformed point clouds in Fig. 9b. It can be seen that after applying input transformation, the distance between different subjects is reduced, which makes our method more robust to hand shape variations.

3) QUALITATIVE RESULTS

Some qualitative results for ICVL, MSRA and NYU datasets are given in Fig. 10. We compare our method with 2D CNN based method Pose-REN [8] and 3D CNN based method [9]. As can be seen in Fig. 10, our method can better leverage the geometric properties of depth data and produces better estimations than 2D CNN or 3D CNN based methods.

V. CONCLUSION

In this paper we propose a novel method for accurate end-to-end 3D hand pose estimation from point sets. To better preserve the geometric properties of depth data, our method directly consumes point sets and predicts the hand poses. We show that by incorporating the semantic information produced by a semantic segmentation network into a hand pose regression network, the performance of hand pose estimation can be improved. To handle the geometric transformations of input point clouds, we propose a new method to transform the point sets into a latent canonical space and inversely transform the predicted hand pose using an inverse transformation matrix. We demonstrate that this can be achieved by learning two transformation matrices and constraining the inverse property of these two matrices. Experiments shows that our method exhibits promising performance on par with state-of-the-arts. We also shows that our method can further improve the performance of hand pose estimation by using fused point clouds from multi-view depth data without any modifications to the network architecture. Future work may focus on designing a new backbone architecture to better leverage the properties of hand point clouds and hand poses.

APPENDIX

The detailed architectures of SegNet and RegNet can be seen in Fig. 11 and Fig. 12.

REFERENCES

[1] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Comput. Vis. Image Understand.*, vol. 108, nos. 1–2, pp. 52–73, Oct./Nov. 2007.

[2] X. Chen, H. Guo, G. Wang, and L. Zhang, "Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2881–2885.

[3] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, Jun./Jul. 2016, pp. 1–9.

[4] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with RGB-D videos and 3D hand pose annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1–11.

[5] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, Feb. 2012.

[6] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik. (2017). "Intel realsense stereoscopic depth cameras." [Online]. Available: https://arxiv.org/abs/1705.05548

[7] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4512–4516.

[8] X. Chen, G. Wang, H. Guo, and C. Zhang. (2017). "Pose guided structured region ensemble network for cascaded hand pose estimation." [Online]. Available: https://arxiv.org/abs/1708.03416

[9] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1991–2000.

[10] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1196–1205.

[11] M. Oberweger and V. Lepetit, "Deepprior++: Improving fast and accurate 3D hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCV)*, vol. 840, Oct. 2017, pp. 585–594.

[12] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 346–361.

[13] C. Choi, S. Kim, and K. Ramani, "Learning hand articulations by hallucinating heat distribution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2017, pp. 3104–3113.

[14] G. Wang, X. Chen, H. Guo, and C. Zhang, "Region ensemble network: Towards good practices for deep 3D hand pose estimation," *J. Vis. Commun. Image Represent.*, vol. 55, pp. 404–414, Aug. 2018.

[15] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1–10.

[16] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3D pose inference from synthetic images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4663–4672.

[17] G. Poier, D. Schinagl, and H. Bischof, "Learning pose specific representations by predicting different views," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 60–69.

[18] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dense 3D regression for hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1–10.

[19] 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," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5079–5088.

[20] S. Baek, K. I. Kim, and T.-K. Kim, "Augmented skeleton space transfer for depth-based hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1–10.

[21] L. Ge, J. Weng, Y. Cai, and J. Yuan, "Hand PointNet: 3D hand pose estimation using point sets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1–10.

[22] S. Yuan *et al.*, "Depth-based 3D hand pose estimation: From current achievements to future goals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2636–2645.

[23] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang. (2017). "Hand3D: Hand pose estimation using 3D neural network." [Online]. Available: https://arxiv.org/abs/1704.02224

[24] S. Yuan, Q. Ye, G. Garcia-Hernando, and T.-K. Kim. (2017). "The 2017 hands in the million challenge on 3D hand pose estimation." [Online]. Available: https://arxiv.org/abs/1707.02237

[25] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5105–5114.

[27] J. S. Supančič, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: Data, methods, and challenges," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1868–1876.

[28] J. S. Supančič, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: Methods, data, and challenges," *Int. J. Comput. Vis.*, to be published, doi: 10.1007/s11263-018-1081-7.

[29] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly, "Robust articulated-ICP for real-time hand tracking," *Comput. Graph. Forum*, vol. 34, no. 5, pp. 101–114, 2015.

[30] T. Sharp *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 3633–3642.

[31] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1106–1113.

[32] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2011, pp. 101.1–101.11.

[33] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3213–3221.

[34] A. Tkach, M. Pauly, and A. Tagliasacchi, "Sphere-meshes for real-time hand modeling and tracking," *ACM Trans. Graph.*, vol. 35, no. 6, 2016, Art. no. 222.

[35] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Trans. Graph.*, vol. 36, no. 6, 2017, Art. no. 245.

[36] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3325–3333.

[37] J. Taylor *et al.*, "Articulated distance fields for ultra-fast tracking of hands interacting," *ACM Trans. Graph.*, vol. 36, no. 6, 2017, Art. no. 244.

[38] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3224–3231.

[39] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3D articulated hand posture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3786–3793.

[40] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 824–832.

[41] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, 2014, Art. no. 169.

[42] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3593–3601.

[43] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," in *Proc. Comput. Vis. Winter Workshop*, 2015, pp. 21–30.

[44] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2421–2427.

[45] J. Malik, A. Elhayek, and D. Stricker. (2017). "Simultaneous hand pose and skeleton bone-lengths estimation from a single depth image." [Online]. Available: https://arxiv.org/abs/1712.03121

[46] E. Dibra, T. Wolf, C. Öztireli, and M. Gross, "How to refine 3D hand pose estimation from unlabelled depth data?" in *Proc. 5th Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 135–144.

[47] Y. Li, R. Bu, M. Sun, and B. Chen. (2018). "PointCNN." [Online]. Available: https://arxiv.org/abs/1801.07791

[48] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1–10.

[49] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. (2018). "Dynamic graph CNN for learning on point clouds." [Online]. Available: https://arxiv.org/abs/1801.07829

[50] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1–10.

[51] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 2017–2025.

[52] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen, "Real-time rotation-invariant face detection with progressive calibration networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2295–2303.

[53] J. Shotton *et al.*, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, Dec. 2013.

[54] M. Sun, P. Kohli, and J. Shotton, "Conditional regression forests for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3394–3401.

[55] N. Neverova, C. Wolf, F. Nebout, and G. W. Taylor, "Hand pose estimation through semi-supervised and weakly-supervised learning," *Comput. Vis. Image Understand.*, vol. 164, pp. 56–67, Nov. 2017.

[56] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[57] D. Fourure *et al.*, "Multi-task, multi-domain learning: Application to semantic segmentation and pose regression," *Neurocomputing*, vol. 251, pp. 68–80, Aug. 2017.

[58] M. Madadi, S. Escalera, A. Carruesco, C. Andujar, X. Baró, and J. Gonzàlez, "Occlusion aware hand pose recovery from sequences of depth images," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May/Jun. 2017, pp. 230–237.

[59] D. Bouchacourt, P. K. Mudigonda, and S. Nowozin, "DISCO Nets: Dissimilarity coefficients networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 352–360.

[60] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3316–3324.

[61] M. Madadi, S. Escalera, X. Baro, and J. Gonzalez. (2017). "End-to-end global to local CNN learning for hand pose recovery in depth data." [Online]. Available: https://arxiv.org/abs/1705.09606

[62] 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," *Int. J. Comput. Vis.*, vol. 123, no. 3, pp. 454–478, 2017.

[63] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

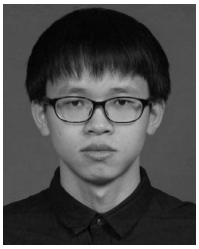
[64] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 6, pp. 914–924, Jun. 2012.

[65] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: An open-source mesh processing tool," in *Proc. Eurograph. Italian Chapter Conf.*, 2008, pp. 129–136.

[66] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

XINGHAO CHEN received the B.S. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree. From 2016 to 2017, he was a Visiting Ph.D. Student with Imperial College London, U.K. His research interests include deep learning, hand pose estimation, and gesture recognition.

GUIJIN WANG received the B.S. and Ph.D. degrees (Hons.) in signal and information processing from the Department of Electronics Engineering, Tsinghua University, China, in 1998 and 2003, respectively. From 2003 to 2006, he was with Sony Information Technologies Laboratories as a Researcher. Since 2006, he has been with the Department of Electronics Engineering, Tsinghua University, as an Associate Professor. In 2012, he was the Visiting Researcher with the AMP Laboratory, Cornell University. He published over 100 international journals and conference papers and holds 10 patents with numerous pending. His research interests focus on computational imaging, pose recognition, intelligent human–machine UI, intelligent surveillance, industry inspection, and AI for big medical data. He received the reward (the first prize) of the Science and Technology Award from the Chinese Association for Artificial Intelligence in 2014 and the reward (the second prize) of Shandong Province Science and Technology Progress in 2014. He was an Associate Editor of the *IEEE Signal Processing Magazine*, the Guest Editor of *Neurocomputing*, the Track Chair of ChinaSIP 2015, and the TPC Member of ICIP2017.



CAIRONG ZHANG received the B.S. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include deep learning, human pose estimation, and hand pose estimation.

TAE-KYUN KIM has been an Associate Professor and the Director of the Computer Vision and Learning Laboratory, Imperial College London, since 2010. On the topics of hand pose, face recognition by image sets, 6-D object pose, action/activity, and robot vision, he has published over 40 top-tier journal and conference papers. His co-authored algorithm is an international standard of MPEG-7 ISO/IEC for face image retrieval, and he was a recipient of the KUKA Best Robotics Paper Award at ICRA14. He has been co-chairing the CVPR HANDS Workshop and the ICCV/ECCV Object Pose workshop. He is the General Chair of BMVC17. He is an Associate Editor of *Image and Vision Computing Journal* and the *IPSJ Transactions on Computer Vision and Applications*.

XIANGYANG JI received the B.S. degree in materials science and the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He joined Tsinghua University, Beijing, in 2008, where he is currently a Professor with the Department of Automation. His current research interests cover signal processing, image/video processing, and machine learning.