IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2019, accepted March 2, 2020, date of publication March 9, 2020, date of current version March 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979507

Light and Fast Hand Pose Estimation From Spatial-Decomposed Latent Heatmap

SHAOWEI LIU, GUIJIN WANG, (Senior Member, IEEE), PENGWEI XIE, AND CAIRONG ZHANG

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China

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

ABSTRACT We present a light and efficient approach named Latent Fusion network for fast and accurate hand pose estimation from a single depth image. Our method innovatively decomposes 3D joint regression into 2D plane localization and 1D axis estimation from different spatial perspectives. We design multiple latent heatmap regression branches to predict hand pose separately and a fusion network to output the final result. Experiments on three public hand pose datasets (ICVL, NYU, MSRA) demonstrate that our system achieves state-of-the-art accuracy. Moreover, our method outperforms all top-ranked approaches by a large margin both in terms of inference speed (nearly a thousand frames per second) and model size (less than 10 MB).

INDEX TERMS Hand pose estimation, convolutional neural network, depth images, heatmap regression, human computer interaction.

I. INTRODUCTION

3D hand pose estimation is the primary technique in human computer interaction and an essential research topic in computer vision community [1]. With the wide availability of all kinds of depth sensors like Intel Realsense [2], Microsoft Kinect [3], depth-based hand pose estimation has attracted much research attention [4]–[8]. Despite the fact that great improvement has been achieved in this field, it's still challenging for accurate and robust estimation in real-time and low-cost. On the one hand, the flexibility of articulated hand pose and severe self-occlusion have made localizing hand joints in 3D space quite difficult. On the other hand, proposed methods should be highly efficient and lightweight to suit real-world applications.

Recent years have witnessed the success of utilizing convolutional neural networks (CNN) in hand pose estimation. CNN-based methods [9]–[18] have an advantage in efficiency and robustness. Given an input depth image, traditional CNN-based methods simply feed it into 2D CNN, while recent studies convert it into 3D voxel beforehand and use 3D CNN to better exploit spatial information. Though greatly improve the estimation accuracy, 3D CNN-based methods suffer from large memory request and low running speed.

The associate editor coordinating the review of this manuscript and approving it for publication was Fanbiao Li.

Joint coordinates and spatial heatmaps are two commonly used representations for the output of CNN-based methods. Coordinates regression-based methods [10], [12]–[14], [17], [19] use fully connected layers to output target joint coordinates directly. Heatmap regression-based methods [11], [15], [20] produce a probability heatmap for each joint in Gaussian distribution, whose peak is positioned at the ground truth joint location and standard deviation σ is manually assigned, representing per-pixel or per-voxel likelihood. Heatmap regression-based methods outperform coordinates regression-based methods in estimation accuracy [8], but it's time-consuming to extract joint locations from heatmaps by non-differentiable argmax operation. Furthermore, explicitly assigned distribution and fixed σ are not ideal for different joints. Therefore, we need to solve the above defects before applying heatmap regression into depth-based hand pose estimation.

In this work, we propose a novel spatial-decomposed latent heatmap regression method with single 2D depth image input and 2D CNN architecture. To tackle the aforementioned problems of heatmap regression, we introduce latent heatmaps in our network. The automatically learnt heatmaps are fully differentiable, and only multiply-accumulate operation is needed to extract joint coordinates. In order to better alleviate the self-occlusion problem and boost estimation performance, we extend our method to multiple regression

branches with different spatial decompositions. The final output is the weighted average of different branches' predictions via a fusion network. Experiments show that our Latent Fusion network has the best overall performance (estimation accuracy, inference speed and model size) on three challenging hand pose datasets (ICVL, NYU, MSRA). Our method outperforms all previous approaches on ICVL dataset and achieves state-of-the-art accuracy on NYU and MSRA dataset. Moreover, our method can run at nearly a thousand frames per second during testing on a single GPU, 8 times faster than the current best result [18] among top-ranked approaches. We also maintain the lightest model size which is less than 10MB.

Our contributions are summarized below.

1) We present a light and fast Latent Fusion network for depth-based hand pose estimation. The network takes a 2D depth image as input and adopts latent heatmap regression to localize 3D hand joints.
2) We design a multi-branch network to regress hand joints from different spatial decompositions. The proposed architecture can better excavate 3D information and alleviate the self-occlusion problem.
3) We propose a novel pose fusion strategy to combine different branches' predictions to give the most accurate final output, which can significantly boost the estimation performance.

The remainder of this paper is organized as follows. Section II reviews related work. Section III introduces the details of our proposed Latent Fusion network. Comprehensive experiments and ablation study are provided in Section IV. Section V gives a conclusion of this paper.

II. RELATED WORK

A. DEPTH-BASED HAND POSE ESTIMATION

Hand Pose estimation from a single depth image can be generally categorized into three classes: generative methods, discriminative methods and hybrid methods. Generative methods [21]–[25] use pre-defined hand models to fit input depth images. Model parameters are optimized by an energy function. Although generative methods leverage kinematic constraints, accumulative estimation error and tedious optimization process have made them impractical for real-world applications. Discriminative methods learn optimal hand joint positions directly. They are data-driven and the most popular way for hand pose estimation. Hybrid methods [26]–[30] combine generative methods and discriminative methods, but still suffer from the same problem as generative methods.

Earlier discriminative methods use random forest [31]–[33] to estimate hand joint locations. They are gradually replaced by CNN-based methods with more powerful feature extractor. CNN-based methods can be divided into two groups: 2D-based and 3D-based. 2D-based CNN methods [10], [12], [14], [17] treat depth image as a single channel image while 3D-based CNN methods transform depth image into voxel [13], [15], [20] or point clouds [19], [34]–[36]. Since it's time-consuming for voxelization in 3D CNN or sampling in point clouds, we adopt 2D CNN in our method without complicated data preprocessing.

B. HEATMAP REGRESSION

Heatmap regression benefits from a stronger supervised signal and fully convolutional network (FCN) architecture. 2D joint heatmap for hand pose estimation was first proposed in [9] and greatly improved by later works. Moon *et al.* [15] adopted 3D heatmap for joint regression. Wan *et al.* [16] utilized both 2D and 3D heatmaps and obtained output by the mean-shift algorithm. However, 2D heatmaps are inadequate for 3D pose estimation, while 3D heatmaps require 3D input. Mixed representations like [16] can't be trained in an end-to-end manner. To this end, we cast hand joint space into multiple heatmap groups from different spatial decompositions to better excavate 3D information and regress hand joint coordinates.

Another challenge to heatmap regression is the selection of σ. If the selected σ is small, the supervised signal will be sparse, and the situation can even degrade to coordinates regression. If the selected σ is large, joint locations will be inaccurate as candidate regions are large. To solve this issue, Wu *et al.* [37] employed dense guidance map with geodesics approximation, but it brought arduous computation. Iqbal *et al.* [38] proposed latent heatmap regression towards RGB images input, which is somewhat inaccurate for depth images with severe self-occlusion. Inspired by this work [38], we combine the idea of latent heatmap regression and spatial decomposition to boost the accuracy for depth-based hand pose estimation.

C. ENSEMBLE AND FUSION

Ensemble and fusion technique has been widely used in hand pose estimation. Moon *et al.* [15] applied epoch ensemble to average pose predictions from different training stage models. Guo *et al.* [12], [17] proposed region ensemble to extract most representative feature regions and fused them for holistic regression. Ge *et al.* [11] projected point clouds of the depth image on different views as multiple inputs and obtained final fusion result by the maximum a posteriori estimation. Different from all previous works, our network consists of multiple separate regression branches with single depth image input. We further design a tiny fusion network to straightly output the weighted average result over different branches' predictions by assigning each prediction a normalized fusion weight.

III. LATENT FUSION NETWORK

The whole architecture of Latent Fusion network is illustrated in Fig. 1. Latent Fusion network takes a 2D depth image as input and outputs corresponding hand joint locations. The main backbone of our network is hourglass module [39] which consists of recursive sampling blocks and residual blocks. The input depth image first goes through a feature extraction network to generate sharing features as input for

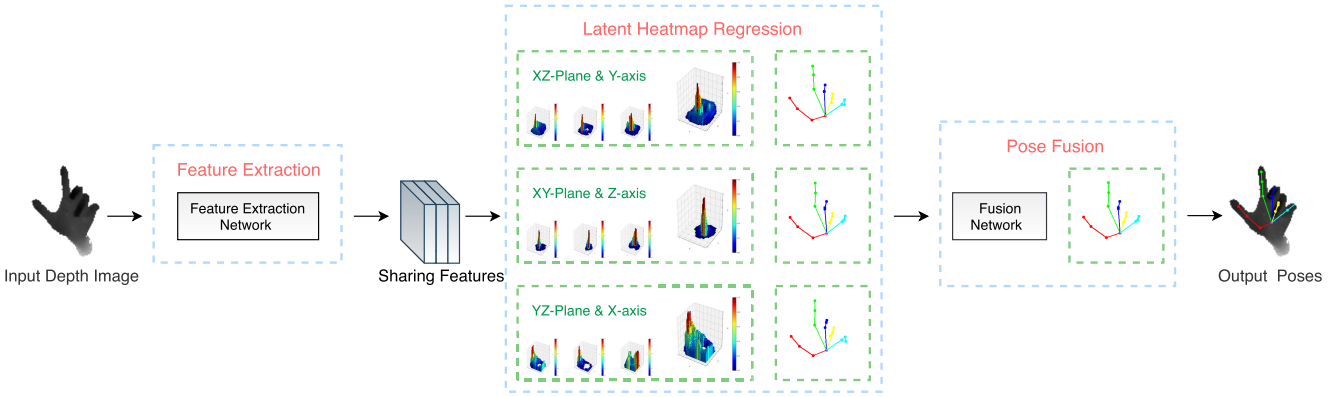

FIGURE 1. Overview of our proposed network architecture. Our model consists of three subnetworks: the feature extraction network, the multi-branch latent heatmap regression network and the pose fusion network. Feature extraction network takes a single 2D depth image as input and generates sharing low-level features for following latent heatmap regression branches. Each branch estimates 3D hand joint coordinates from a group of latent heatmaps with different spatial decomposition. The final output is the weighted average of all three branches' predictions via the fusion network. The whole framework is trained in an end-to-end manner by introducing a separate loss function for each branch and the fusion output.

different regression branches. Each branch conducts latent heatmap regression from different spatial decomposition separately. The final pose is the weighted average of all branches' predictions via the fusion network.

We describe details in the following order. III-A introduces latent heatmap regression. III-B extends it to multi-branch prediction with spatial decomposition. III-C describes the fusion network. III-D presents the loss function. Implementation details are discussed in III-E.

A. LATENT HEATMAP REGRESSION

The goal for 3D hand pose estimation is to learn a mapping function between input depth image I and output hand joint coordinates $J = \{p_j\}_{j=1}^{M} \in R^{3\times M}$, where $p_j = (u_j, v_j, d_j)$ is the 3D location of joint j and M is the total number of joints. Instead of using explicit heatmap regression with fixed σ to generate Gaussian distribution, we adopt latent heatmap regression proposed in [38] to learn an optimal distribution automatically without constraints. In order to legalize the learnt distribution, spatial softmax normalization is applied on the latent heatmap to enforce its summation equals to one strictly. Since we use latent heatmaps in 2D form, so normalization is operated on each joint j's 2D heatmap plane as follows:

$$H_j^*(u, v) = \frac{exp(H_j(u, v))}{\sum_{u'} \sum_{v'} exp(H_j(u', v'))}. \tag{1}$$

where u and v are pixel locations in the 2D plane, H_j and H_j^* are latent 2D heatmaps before and after spatial normalization. Since latent 2D heatmaps only provide per-pixel likelihood and depth information is needed for localizing hand joints in 3D space, we output latent 1D heatmaps for depth estimation along with latent 2D heatmaps. Each element in latent 1D heatmap D_j is a predicted depth value of the corresponding position at normalized latent 2D heatmap H_j^*. We can obtain 3D coordinates of joint j via H_j^* and D_j by the following

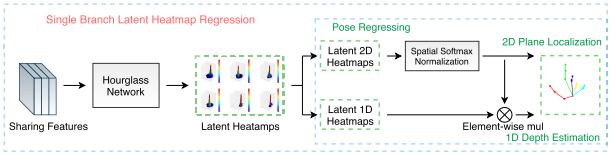

FIGURE 2. The detailed architecture of latent heatmap regression. As 3D joint regression is decomposed into 2D plane localization and 1D depth estimation, the network outputs a latent 2D heatmap and a latent 1D heatmap for each joint separately. After applying spatial softmax normalization on latent 2D heatmaps, output pose prediction can be obtained by equation 2.

equations:

$$u_j = \sum_{u'} \sum_{v'} H_j^*(u', v') \cdot \widetilde{u'} \tag{2a}$$

$$v_j = \sum_{u'} \sum_{v'} H_j^*(u', v') \cdot \widetilde{v'} \tag{2b}$$

$$d_j = \sum_{u'} \sum_{v'} D_j(u', v') \cdot H_j^*(u', v') \tag{2c}$$

where $\widetilde{u'}$ and $\widetilde{v'}$ are corresponding normalized pixel coordinates within $[-1, 1]$. Pixel location (u_j, v_j) is the centroid of normalized latent 2D heatmap H_j^* while depth estimation d_j is the weighted average of latent 1D heatmap D_j and normalized latent 2D heatmap H_j^*. The detailed architecture of latent heatmap regression is shown in Fig. 2. Our network outputs $2M$ feature maps simultaneously with M latent 2D heatmaps and M latent 1D heatmaps. Therefore, after applying spatial softmax normalization on latent 2D heatmaps, by using equation 2 and multiply-accumulate operation, we can obtain joint coordinates $\{p_j\}_{j=1}^{M}$ easily.

B. SPATIAL DECOMPOSITION

Section III-A introduces single branch latent heatmap regression, as normalized latent 2D heatmap H_j^* provides per-pixel likelihood in XY-plane while latent 1D heatmaps predict corresponding depth value along the Z-axis. In this way,

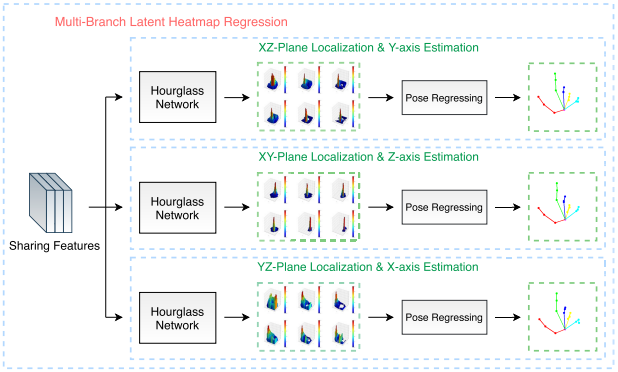

FIGURE 3. Illustration of the multi-branch latent heatmap regression. The middle branch decomposes 3D joint regression into XY-plane localization and Z-axis estimation, while the top and the bottom are two additional regression branches with different spatial decompositions. The multi-branch network can better excavate 3D information and provide more accurate joint estimation than single branch regression.

3D joint regression is decomposed into 2D plane localization (XY-plane) and 1D axis estimation (Z-axis). Since single branch regression is insufficient for accurate hand pose estimation under some extreme cases, spatial decomposition strategy can learn richer 3D representation and better alleviate the self-occlusion problem. Specifically, we design three separate latent heatmap regression branches and cyclically decompose X, Y, Z three-axis into 2D plane localization and 1D axis estimation inside each branch. Hence the multi-branch regression network contains three different spatial decompositions (XY-plane and Z-axis, XZ-plane and Y-axis, YZ-plane and X-axis). Each regression branch has the same architecture and receives common sharing input features, as shown in Fig. 3. Besides, considering the fact that the spread of latent 2D heatmaps for the same joint should be identical no matter how spatial perspective changes, we introduce a group of sharing parameters ω as cross-branch constraint for different joints. The learnable group of parameters ω control the spread of heatmaps and are updated synchronously among all branches. Thus, the spatial normalization for joint j in equation 1 can be reformulated as follows:

$$H_j^*(u,v) = \frac{exp(\omega_j H_j(u,v))}{\sum_{u'}\sum_{v'} exp(\omega_j H_j(u',v'))}. \tag{3}$$

where ω_j is the sharing parameter of joint j in all three branches. All other operations inside each branch remain the same as section III-A.

C. FUSION NETWORK

As each branch gives an independent estimation $\{p_j\}_{j=1}^M$, the final output should be the aggregation of all predictions. Therefore, we assign a fusion weight for each branch's prediction. The fusion weights are produced by the fusion network which receives transformation input from sharing features and latent heatmaps illustrated in Fig. 4. Inside the network, we use two convolutional layers and two max pooling layers to reduce the spatial size of feature maps. In order to generate a 1×1 fusion weight for each prediction, we employ global average pooling [40] and sigmoid activation function at the end of the network. The final fusion result of joint j is obtained as follows:

$$p_j^f = \sum_{i=1}^{3} \mu_i p_j^i \tag{4}$$

where p_j^i denotes the pose prediction from branch i and μ_i represents the corresponding normalized fusion weight.

D. LOSS FUNCTION

The loss function for branch i is defined as follows:

$$L_i = L_{sml1}(J^i, J) \tag{5}$$

where $J^i = \{p_j^i\}_{j=1}^M$ denotes the predicted hand joint coordinates and J is ground truth labels. L_{sml1} is the smooth L_1 loss

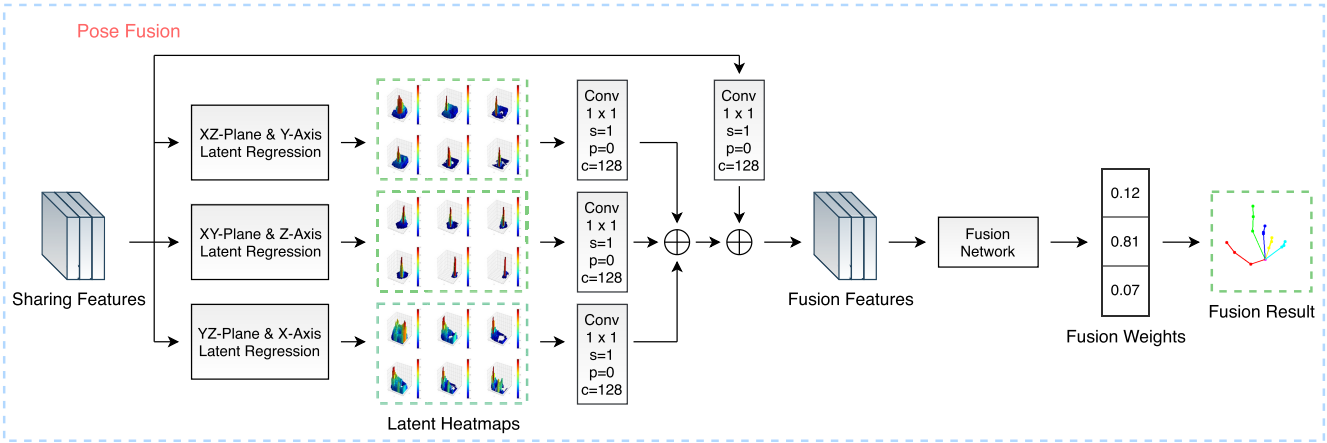

FIGURE 4. The detailed architecture of pose fusion network. Sharing features and latent heatmaps (including M normalized latent 2D heatmaps H_j^* and M latent 1D heatmaps D_j in each branch) go through 1×1 convolutional layers separately and sum up together as fusion features for the fusion network. The fusion network produces a normalized fusion weight for each branch's prediction. The final fusion result can be obtained by equation 4.

proposed in [44]. Loss function L_f for the final fusion result is defined in a similar way.

The total loss function for the entire network is:

$$L = L_f + \sum_{i=1}^{3} L_i \tag{6}$$

E. IMPLEMENTATION DETAILS

Our network is implemented in Pytorch [45]. The detailed architecture of feature extraction network, hourglass network and fusion network can be found in Table 6 in appendix. The GeForce GTX 1080 Ti GPU is used for training and testing.

1) PREPROCESSING

We follow the similar strategy as prior works [10], [12], [14], [17] to extract a fixed-size cube of original depth image and resize it to a 128×128 patch as input. Depth values within the cropped region and corresponding poses are normalized into $[-1, 1]$. We also adopt random translation ($[-10, 10]$ pixel), scaling ($[0.9, 1.1]$) and rotation ($[-15, 15]$ degrees along z-axis) for online data augmentation during training.

2) PARAMETER SETTINGS

We choose 32×32 as the input and output resolution of hourglass module and fix feature channels (128) in each block. All network weights are initialized from zero-mean Gaussian distribution with $\sigma = 0.001$. Learnable parameters ω for controlling the spread of latent heatmaps are initialized with 1.

3) TRAINING AND TESTING

We use Adam [46] optimizer to train the network for 100 epochs with batch size of 32. The learning rate is set to 0.001 and divided by 10 every 30 epochs. During testing, our network can achieve 946 fps on a single GPU.

IV. EXPERIMENTS

A. DATASETS AND EVALUATION METRICS

1) ICVL DATASET

ICVL dataset [32] contains 330K frames for training and 1596 samples for testing. Each frame is labeled with 16 joints, including 3 joints (Root, Middle, Tip) per finger and 1 joint for the palm.

2) NYU DATASET

NYU dataset [9] contains 72757 frames for training and 8252 samples for testing. Since one subject in test set doesn't appear in training set and large hand poses have been covered, the dataset is quite challenging and far from saturation. Following the protocol from previous works, we use 14 joints from the frontal view out of 36 annotated joints for evaluation.

3) MSRA DATASET

MSRA dataset [33] contains 76K frames from 9 different subjects. The leave-one-subject-out cross-validation strategy is utilized for evaluation. Each depth image is labeled with 21 joints, including 4 joints (MCP, PIP, DIP, TIP) per finger and 1 joint for the palm.

4) EVALUATION METRICS

We adopt two most commonly used metrics for evaluation: mean joint error and success rate. The former is the average Euclidean distance between predicted joint coordinates and annotated ones, while the latter is the proportion of test frames whose all joint errors fall below a threshold.

B. COMPARISON WITH STATE-OF-THE-ARTS

To evaluate the overall performance (estimation accuracy, inference speed and model size), we compare estimation accuracy against most state-of-the-art methods on ICVL [32], NYU [9] and MSRA [33] datasets. Among those top-ranked approaches on three datasets, we further compare the inference speed and model size to demonstrate the superiority of our method.

1) ESTIMATION ACCURACY

Comparison methods towards estimation accuracy include latent regression forest (LRF) [32], hands deep with pose prior (DeepPrior) [10], improved DeepPrior (DeepPrior++) [14], feedback loop training (Feedback) [28], cascaded hand pose regression (Cascaded) [33], model-based hand pose estimation (DeepModel) [30], Lie group-based method (Lie-X) [43], multi-view 2D CNNs (Multiview) [11], joint training with shared context method (JTSC) [41], region ensemble network (REN-4 $\times$ 6 $\times$ 6 [12], REN-9 $\times$ 6 $\times$ 6 [42]), pose guided structure REN (Pose-REN) [17], dense 3D Regression (DenseReg) [16], 3DCNN [13], Voxel-to-Voxel Prediction Network (V2V-PoseNet) [15], HandPointNet [34], Point-to-Point Regression PointNet (Point-to-Point) [35], SHPR-Net [19] and CrossInfoNet [18].

On ICVL dataset, we compare our method against [10], [12], [14]–[19], [30], [32], [34], [35], [41], [42]. As shown in Fig. 5 and Table 1a, we exceed all state-of-the-art methods above by a large margin. Specifically, our method has the lowest mean joint error of 5.81 mm, achieving 7.5% relative improvement compared to the previous best method [15].

On NYU dataset, we compare with [10], [12]–[19], [28], [30], [34], [35], [41]–[43]. As can be seen from Fig. 6 and Table 1b, our method outperforms most state-of-the-art methods and is on par with the rest of them. When the error threshold is less than 8 mm, our method achieves the best performance among all evaluated methods. However, with the growing of maximum allowed error threshold, our method's performance drops slightly, which can be attributed to the simple hand segmentation strategy used in III-E.1. As NYU dataset [9] is captured by structure light camera and contains many invalid pixels, naive depth thresholding and fixed-size cube can't filter noisy background completely. Other methods like [15] design additional networks to refine hand localizations and extract hand regions.

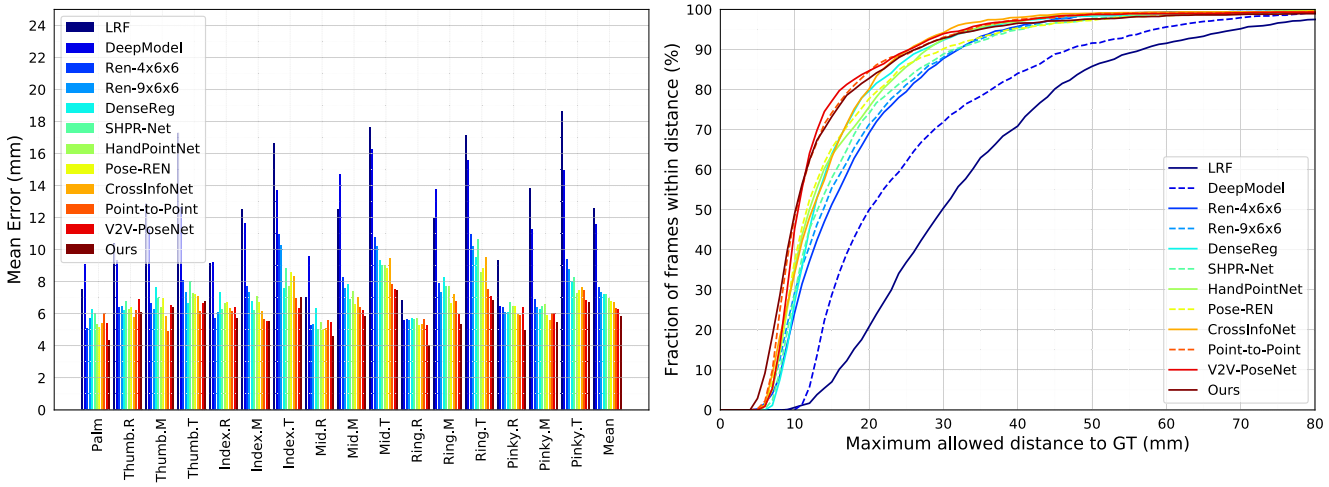

FIGURE 5. Comparison with state-of-the-art methods on ICVL dataset. Left: mean errors per-joint. Right: the percentage of success frames over different error thresholds.

TABLE 1. Comparison of proposed method (Latent Fusion) with previous state-of-the-art methods on three hand pose datasets. Error indicates the average 3D distance error over all joints.

Methods	Error (mm)
LRF [32]	12.58
DeepModel [30]	11.56
DeepPrior [10]	10.4
JTSC [41]	9.16
DeepPrior++ [14]	8.1
REN (4x6x6) [12]	7.63
REN (9x6x6) [42]	7.31
DenseReg [16]	7.3
SHPR-Net [19]	7.22
HandPointNet [34]	6.9
Pose-REN [17]	6.79
CrossInfoNet [18]	6.73
Point-to-Point [35]	6.3
V2V-PoseNet [15]	6.28
Latent Fusion (Ours)	**5.81**

(a) ICVL

Methods	Error (mm)
DeepPrior [10]	19.73
DeepModel [30]	16.90
JTSC [41]	16.80
Feedback [28]	15.97
Lie-X [43]	14.51
3DCNN [13]	14.1
REN (4x6x6) [12]	13.39
REN (9x6x6) [42]	12.69
DeepPrior++ [14]	12.24
Pose-REN [17]	11.81
SHPR-Net [19]	10.78
HandPointNet [34]	10.5
DenseReg [16]	10.21
CrossInfoNet [18]	10.08
Point-to-Point [35]	9.1
V2V-PoseNet [15]	**8.42**
Latent Fusion (Ours)	9.69

(b) NYU

Methods	Error (mm)
Cascaded [33]	15.2
Multi-view CNNs [11]	13.1
REN (9x6x6) [42]	9.79
3DCNN [13]	9.58
DeepPrior++ [14]	9.5
Pose-REN [17]	8.65
HandPointNet [34]	8.5
CrossInfoNet [18]	7.86
SHPR-Net [19]	7.76
Point-to-Point [35]	7.7
V2V-PoseNet [15]	7.59
DenseReg [16]	**7.2**
Latent Fusion (Ours)	7.90

(c) MSRA

On MSRA dataset, we compare the performance of our method with [11], [13]–[19], [33]–[35], [42]. Results are shown in Fig. 7 and Table 1c. As illustrated, we get comparable result with [15], [18], [19], [35]. Though inferior to [16], our method gains an absolute edge in inference speed and model size.

Qualitative results for ICVL, NYU and MSRA datasets are shown in Fig. 9, Fig. 10, Fig. 11 respectively. As can be seen, our method can well capture complex hand structures with different joint annotations and effectively alleviate the self-occlusion problem.

2) INFERENCE SPEED AND MODEL SIZE

The runtime of our method is 1.057ms per frame on average, including 0.145ms for feature extraction, 0.168ms for latent heatmap regression, 0.020ms for pose fusion and 0.724ms for post-processing. Thus, our method runs at 946 fps on a single GPU. We compare the inference speed with [15], [16], [18], [34], [35] who have top estimation accuracy on three hand pose datasets. As Table 2 listed, our method is about 270 times, 34 times, 23 times, 20 times, 8 times faster than [15], [16], [34], [35] and [18] respectively. This justifies our method's outstanding real-time performance.

In addition, our model has only 2.000M number of parameters, including 0.510M for the feature extraction network, 1.419M for the multi-branch latent heatmap regression network and 0.071M for the pose fusion network. The model size of our method is 7.9MB. We compare the model size against [15], [16], [18], [34], [35]. As shown in Table 3, our method lies in the first place, light enough for real-world applications. The detailed runtime and model parameters profile can be seen in Table 4.

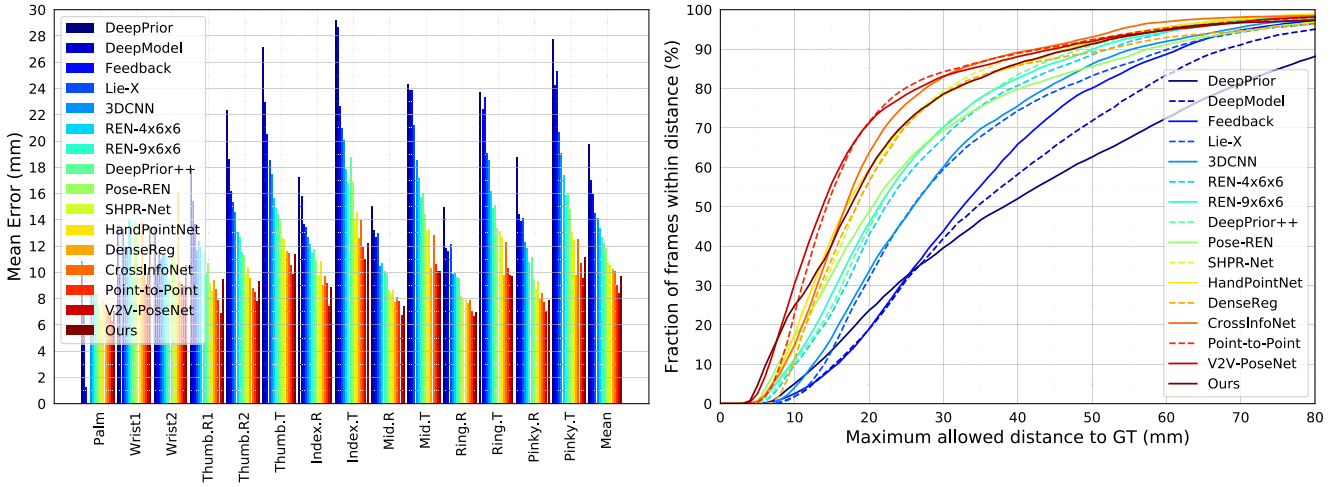

FIGURE 6. Comparison with state-of-the-art methods on NYU dataset. Left: mean errors per-joint. Right: the percentage of success frames over different error thresholds.

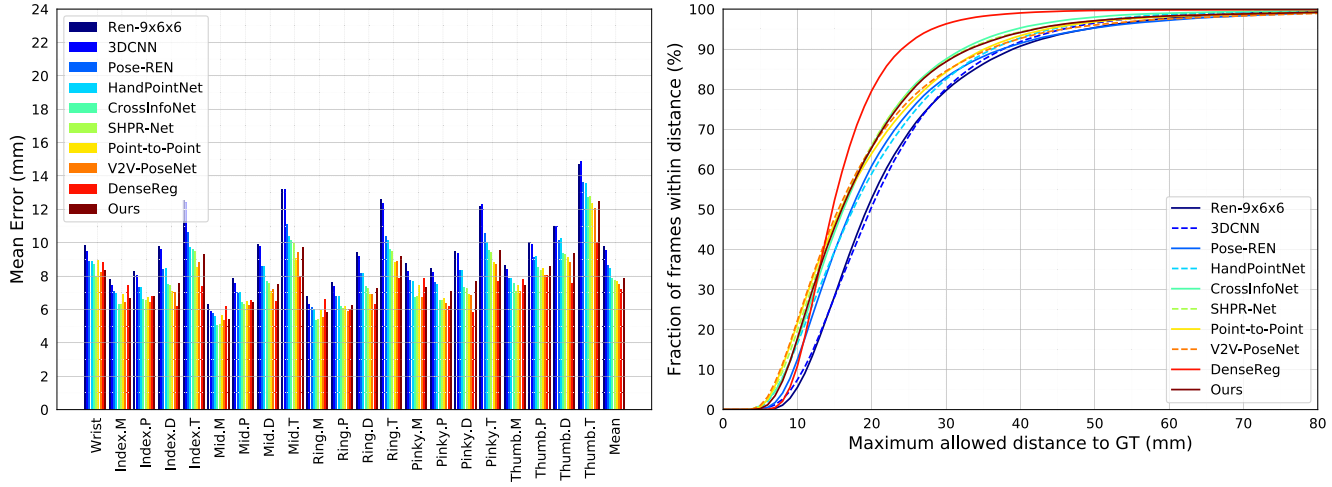

FIGURE 7. Comparison with state-of-the-art methods on MSRA dataset. Left: mean errors per-joint. Right: the percentage of success frames over different error thresholds.

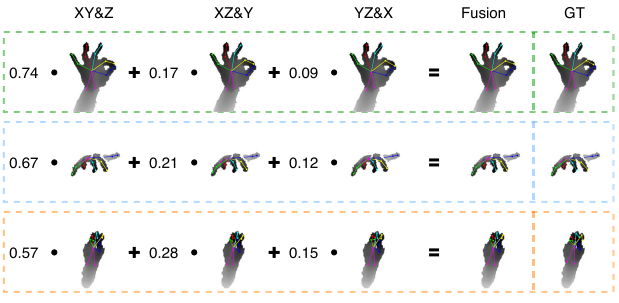

FIGURE 8. Visualization of pose predictions from different regression branches as well as fusion output on NYU dataset. The first three columns are predictions from three branches with different spatial decompositions, i.e., XY-plane and Z-axis, XZ-plane and Y-axis, YZ-plane and X-axis. The number on the left is the normalized fusion weight. The last two columns present fusion results and ground truth hand joint locations.

C. ABLATION STUDY

In this section, we conduct ablation experiments on NYU dataset to analyze the impacts of different module design in Latent Fusion network. We incrementally introduce five baselines for comparison: 1) Explicit Heatmap (B1). We adopt single branch explicit heatmap regression. The target heatmap is a 2D Gaussian centred at the ground truth joint position with fixed $\sigma = 1.5$. 2) Latent Single YZ&X (B2). We adopt single branch latent heatmap regression with YZ-plane localization and X-axis estimation. 3) Latent Single XZ&Y (B3). We adopt single branch latent heatmap regression with XZ-plane localization and Y-axis estimation. 4) Latent Single XY&Z (B4). We adopt single branch latent heatmap regression with XY-plane localization and Z-axis estimation. 5) Latent Ensemble (B5). We adopt spatial-decomposed multi-branch latent heatmap regression, as discussed in section III-B, but substitute ensemble technique for fusion strategy. The final result is obtained by averaging all branches' predictions.

For a fair comparison, we use two stacks hourglass module in baseline 1, 2, 3, 4 to ensure a similar number of parameters against the other ones. Other network settings

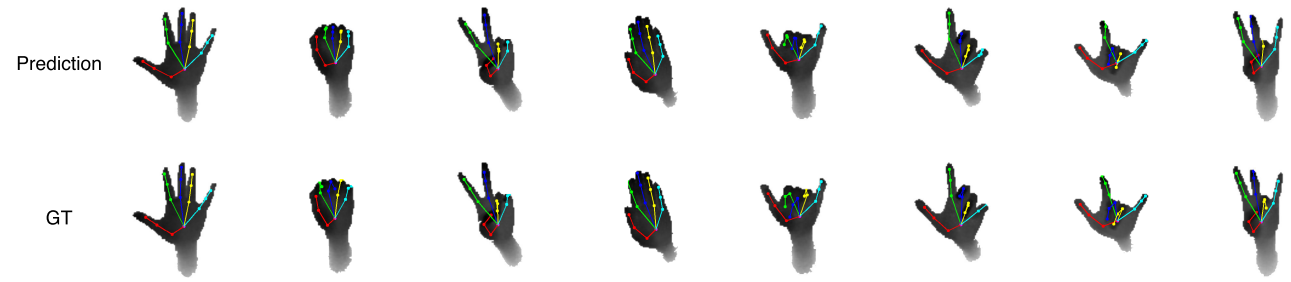

FIGURE 9. Qualitative results on ICVL dataset. The first row is our model predictions while the second row presents corresponding ground truth hand joint locations.

FIGURE 10. Qualitative results on NYU dataset. The first row is our model predictions while the second row presents corresponding ground truth hand joint locations.

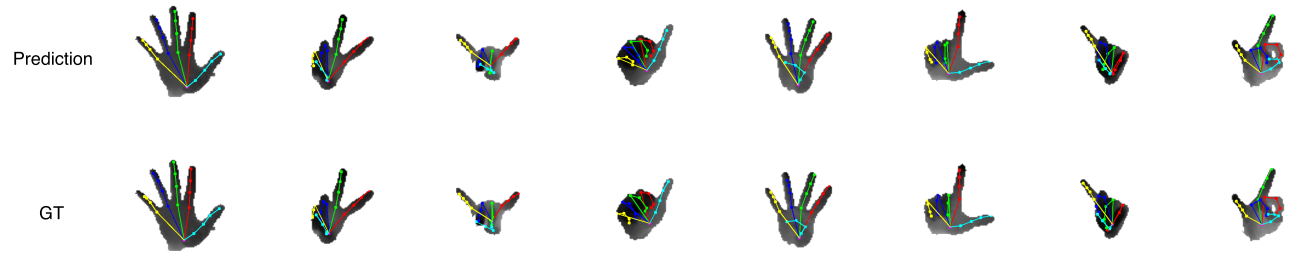

FIGURE 11. Qualitative results on MSRA dataset. The first row is our model predictions while the second row presents corresponding ground truth hand joint locations.

TABLE 2. Comparison of inference speed during testing on a single GPU.

Methods	Speed (fps)
CrossInfoNet [18]	124.5
HandPointNet [34]	48
Point-to-Point [35]	41.8
DenseReg [16]	27.8
V2V-PoseNet [15]	3.5
Latent Fusion (Ours)	**946**

TABLE 3. Comparison of model size.

Methods	Model Size (MB)
CrossInfoNet [18]	95.1
DenseReg [16]	84
V2V-PoseNet [15]	27
Point-to-Point [35]	17.2
HandPointNet [34]	10.3
Latent Fusion (Ours)	**7.9**

TABLE 4. Runtime and model parameters profile.

Modules	Runtime (ms)	Parameters (M)
Feature Extraction	0.145	0.510
Latent Heatmap Regression	0.168	1.419
Pose Fusion	0.020	0.071
Post-Processing	0.724	None
Total	**1.057**	**2.000**

TABLE 5. Comparison with five baselines regarding mean joint error on NYU dataset.

Baselines	Error (mm)
Explicit Heatmap (B1)	14.87
Latent Single YZ&X (B2)	12.48
Latent Single XZ&Y (B3)	11.78
Latent Single XY&Z (B4)	10.47
Latent Ensemble (B5)	9.99
Latent Fusion (Proposed)	**9.69**

remain the same. As shown in Table 5, latent heatmap regression reduces the mean joint error by a large margin against the explicit one. Among three single latent heatmap regression branches, spatial decomposition of XY-plane and Z-axis

TABLE 6. The detailed architecture of feature extraction network, hourglass network and fusion network. The abbreviations N, K, S, P stand for output channels, kernel size, stride and padding respectively.

Feature Extraction Network	
1	Conv-(N16,K7,S2,P3), BatchNorm, LeakyRelu
2	Residual Block-(N32)
3	Max Pooling-(K2,S2)
4	Residual Block-(N64)
5	Residual Block-(N128)
6	2nd-Hourglass Module
7	Residual Block-(N128)
Hourglass Network	
1	2nd-Hourglass Module
2	Residual Block-(N128)
3	Conv-(N128,K1,S1), BatchNorm, LeakyRelu
4	Conv-(N2M,K1,S1)
Fusion Network	
1	Conv-(N32,K1,S1)
2	Max Pooling-(K2,S2)
3	BatchNorm, LeakyRelu
4	Conv-(N3,K1,S1)
5	Global Average Pooling
6	Sigmoid Activation
7	L1 Norm

has the lowest mean error, since it better leverages the spatial geometry of input depth image. However, together with two additional branches by either ensemble or fusion strategy, it can significantly boost the estimation accuracy. The multi-branch network design can better excavate spatial correlations inside the depth image and alleviate the self-occlusion problem. Moreover, the fusion strategy is superior to ensemble one and contributes 0.3 mm accuracy on NYU dataset. As the ensemble technique treats all branches with equal importance, those inaccurate results in three branches can drop the estimation performance. Hence fusion strategy is more suitable for spatial-decomposed multi-branch network and can output the most accurate hand poses.

The visualization of pose predictions from different regression branches as well as fusion output can be seen in Fig. 8. As illustrated, the regression branch of XY-plane localization and Z-axis estimation has high weight under no occlusion circumstances, while the other two branches share more weights as the occlusion in the input depth image becomes severe.

V. CONCLUSION

In this paper, we propose a novel Latent Fusion network for depth-based hand pose estimation. Our method utilizes spatial-decomposed latent heatmaps to estimate hand joint coordinates inside multiple regression branches. We also design a fusion network for aggregating all branches' predictions to output the final hand pose. Experimental results show that our method achieves the best overall performance (estimation accuracy, inference speed and model size) among state-of-the-art approaches. Not only does our method own top accuracy on three public hand pose datasets, but our method can run supremely fast at 946 fps while maintaining the lightest model size of 7.9MB. Possible future work includes hand pose estimation when interacting with objects as well as jointly estimate of hand pose and shape in the wild.

APPENDIX

The detailed architecture of feature extraction network, hourglass network and fusion network can be seen in Table 6.

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. 2007.
[2] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel(R) RealSense(TM) stereoscopic depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1–10.
[3] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultimediaMag.*, vol. 19, no. 2, pp. 4–10, Feb. 2012.
[4] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using kinect," in *Proc. Brit. Mach. Vis. Conf.*, 2011, vol. 1, no. 2, p. 3.
[5] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, Dec. 2013.
[6] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Consumer Depth Cameras for Computer Vision*. London, U.K.: Springer, 2013, pp. 119–137.
[7] J. S. Supancic, 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.
[8] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, J. Yuan, X. Chen, G. Wang, F. Yang, K. Akiyama, Y. Wu, Q. Wan, M. Madadi, S. Escalera, S. Li, D. Lee, I. Oikonomidis, A. Argyros, and T.-K. Kim,, "Depth-based 3D hand pose estimation: From current achievements to future goals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2636–2645.
[9] 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, pp. 1–10, Sep. 2014.
[10] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," in *Proc. Comput. Vis. Winter Workshop*, 2015, pp. 1–10.
[11] 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.
[12] 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.
[13] 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.
[14] M. Oberweger and V. Lepetit, "DeepPrior++: Improving fast and accurate 3D hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 585–594.
[15] J. Y. Chang, G. Moon, 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/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.
[16] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Dense 3D regression for hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5147–5156.
[17] X. Chen, G. Wang, H. Guo, and C. Zhang, "Pose guided structured region ensemble network for cascaded hand pose estimation," *Neurocomputing*, to be published.
[18] K. Du, X. Lin, Y. Sun, and X. Ma, "CrossInfoNet: Multi-task information sharing based hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9896–9905.

[19] 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. 43425–43439, 2018.
[20] F. Huang, A. Zeng, M. Liu, J. Qin, and Q. Xu, "Structure-aware 3D hourglass network for hand pose estimation from single depth image," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–12.
[21] 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.*, Jun. 2014, pp. 1106–1113.
[22] 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, Aug. 2015.
[23] A. Tkach, M. Pauly, and A. Tagliasacchi, "Sphere-meshes for real-time hand modeling and tracking," *ACM Trans. Graph.*, vol. 35, no. 6, p. 222, Nov. 2016.
[24] J. Taylor, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, J. Shotton, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, and J. Valentin, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, Jul. 2016.
[25] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Trans. Graph.*, vol. 36, no. 6, p. 245, 2017.
[26] P. Krejov, A. Gilbert, and R. Bowden, "Combining discriminative and model based approaches for hand pose estimation," in *Proc. 11th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit. (FG)*, vol. 1, May 2015, pp. 1–7.
[27] 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.
[28] 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.
[29] 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.* Cham, Switzerland: Springer, 2016, pp. 346–361.
[30] 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.
[31] 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.*, Dec. 2013, pp. 3224–3231.
[32] 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.*, Jun. 2014, pp. 3786–3793.
[33] 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.
[34] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet: 3D hand pose estimation using point sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8417–8426.
[35] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 475–491.
[36] S. Li and D. Lee, "Point-to-pose voting based hand pose estimation using residual permutation equivariant layer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11927–11936.
[37] X. Wu, D. Finnegan, E. O'Neill, and Y.-L. Yang, "HandMap: Robust hand pose estimation via intermediate dense guidance map supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 237–253.
[38] U. Iqbal, P. Molchanov, T. B. J. Gall, and J. Kautz, "Hand pose estimation via latent 2.5 D heatmap regression," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 118–134.
[39] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 483–499.
[40] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–10.
[41] D. Fourure, R. Emonet, E. Fromont, D. Muselet, N. Neverova, A. Trémeau, and C. Wolf, "Multi-task, multi-domain learning: Application to semantic segmentation and pose regression," *Neurocomputing*, vol. 251, pp. 68–80, Aug. 2017.
[42] 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.
[43] 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, Jul. 2017.
[44] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[45] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS-W*, 2017.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

SHAOWEI LIU received the B.S. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2019. He is currently pursuing the M.S. degree with the University of California San Diego. His research interests include fine-grained image classification, hand pose estimation, and image quality assessment.

GUIJIN WANG (Senior Member, IEEE) received the B.S. and Ph.D. degrees (Hons.) from the Department of Electronics Engineering, Tsinghua University, China, in 1998 and 2003, respectively, in signal and information processing. From 2003 to 2006, he was with Sony Information Technologies Laboratories as a Researcher. Since October 2006, he has been with the Department of Electronics Engineering, Tsinghua University, as an Associate Professor. From January 2012 to June 2012, he was a Visiting Researcher with the AMP Lab, Cornell. He has published more than 100 International journals and conference papers. He holds tens of 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 was a recipient of the reward (the first prize) of Science and Technology Award of 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*, a Guest Editor of *Neurocomputing*, the Track Chair of ChinaSIP 2015, and a TPC member of ICIP2017.

PENGWEI XIE received the B.S. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include deep learning, hand pose estimation, and gesture recognition.

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.