Mining Multi-View Information: A Strong Self-Supervised Framework for Depth-based 3D Hand Pose and Mesh Estimation

Pengfei Ren, Haifeng Sun, Jiachang Hao, Jingyu Wang, Qi Qi, Jianxin Liao
Beijing University of Posts and Telecommunications
rpf, hfsun, haojc, wangjingyu, qiqi8266@bupt.edu.cn; jxlbupt@gmail.com

In the supplemental material, we provide

- the details of single-view estimation network in Sec. 1,
- the details of cross-view fusion network in Sec. 2,
- more details of group-wise confidence encoding in Sec. 3,
- the details of prior loss in Sec. 4,
- more details of training process in Sec. 5,
- the impact of the training data in Sec. 6,
- more qualitative results in Sec. 7.

Note that all the notation and abbreviations here are consistent with the main manuscript.

1. Single-View Estimation Network

Our single-view estimation network adopts a encoder-decoder structure similar to Hourglass network [8]. The difference is that we replace the encoder with ResNet-18 [3] to extract more representative global features for hand model estimation. The target of the decoder is to extract the 2D feature maps $\mathbf{F} \in \mathbb{R}^{256\times32\times32}$, from which we can obtain the intermediate representations $\mathbf{H} \in \mathbb{R}^{21\times32\times32}$, $\mathbf{D} \in \mathbb{R}^{(21\times3)\times32\times32}$ and $\mathbf{W} \in \mathbb{R}^{21\times32\times32}$. For the two-stacked network, we adopt the stacking method in [8] to feed the intermediate representations and visual features of the first network into the second network to obtain more accurate and robust results.

2. Cross-View Fusion Network

In this section, we outline the details of the cross-view fusion network. As shown in Fig. 1, we adopt an encoder-decoder structure to capture multi-scale dependencies between nodes. In particular, we use the average pooling to cluster the skeletal nodes, which shows better performance than the maximum pooling [1, 11] in our experiments. For graph convolution, in order to enhance the flexibility of information passing, we learn individual weights between node pairs in different layers as proposed in SemGCN [13]. Meanwhile, we apply different parameter matrices to perform feature transformations of each node feature as proposed by Liu *et al*. [6], which can learn diverse relational patterns between different hand joints. Formally, given a node feature $\mathbf{x}_i \in \mathbb{R}^Z$, the graph convolution operation can be written as:

$$\mathbf{x}_i^{'} = ReLU(\sum_{k\in\mathcal{N}_i} \mathbf{W_k}\mathbf{x}_k a_{ik}), \tag{1}$$

where $\mathcal{N}_i$ denotes the neighbors of node i including the node itself; $\mathbf{W}_k \in \mathbb{R}^{Z\times Z}$ is a learnable matrix used to transform output channels of $\mathbf{x}_k$; a_{ik} represents the learnable connection weight between the node i and node k.

3. Group-wise Confidence Encoding

Since we do not supervise the weight map during training, the weight map tends to be adaptively distributed to joint-related regions. Therefore, weight maps of different joints have different distributions. As shown in Fig. 2, the weight map of the fingertip tends to concentrate in a small area and presents a 2D Gaussian distribution, while the weight map of the joint in the middle of the finger tends to show a stripe distribution along with the finger. Thus, there is a significant difference in the interval of the maximum distribution of each joint. To solve this problem, as shown in Fig. 3 (a), we divide the joints of different fingers at the same level into the same group. We concatenate the coordinate and confidence of the same joint together and perform feature embedding by grouping-specific FC layers.

4. Prior Loss

The prior loss consists of a shape loss term and a collision loss term.

$$L_{prior} = w_{shape}L_{shape} + w_{coll}L_{coll}, \tag{2}$$

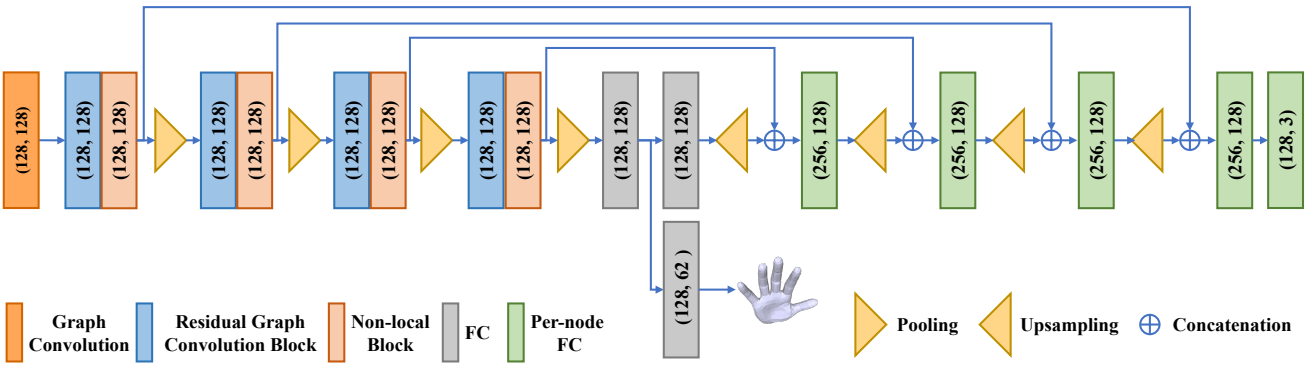

Figure 1. The structure of the coress-fusion network. Similar to previous work [13], the residual graph convolutional block is built by two graph convolutions with a residual skip connection. The non-local block [13] is introduced to facilitate global information passing. The numbers in each module indicate the number of input and output channels respectively.

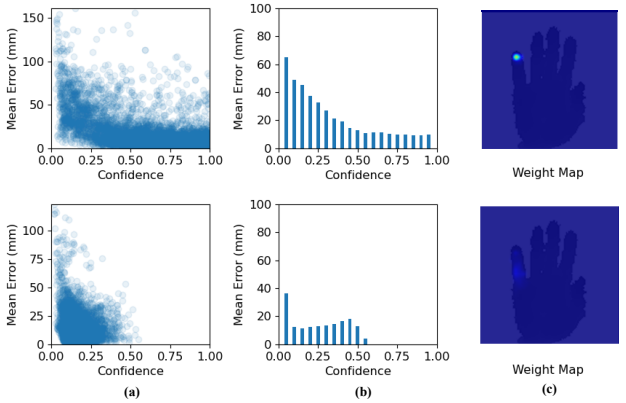

Figure 2. (a) Scatter of the relationship between confidence and mean error. (b) Histogram of the relationship between confidence and error. Specifically, we first partition the [0, 1] range of confidence into 20 equal bins and then we group the samples into corresponding bins according to their confidence. After that we calculate the average error of each bin. (c) Weight map normalized by $softmax$.

where w_{shape} and w_{coll} are constant weights. The shape loss L_{shape} constrains the predicted hand shape β as close as possible to the average shape $\hat{\beta} = \overrightarrow{0}$. The shape loss term is defined as:

$$L_{shape} = \left\| \beta - \hat{\beta} \right\|_2^2. \quad (3)$$

The collision loss L_{coll} is achieved by placing multiple spheres in the hand model and penalizing overlaps between these spheres. Specifically, for the fingers, we trisect each bone evenly and get ten keypoints for each finger. For the palm, we quarter each bone evenly and get five keypoints for each bone. Excluding the repetitive keypoints, there are 66 keypoints remaining, which serve as centers of the spheres. We show the sphere centers in Fig. 3 (b). L_{coll}

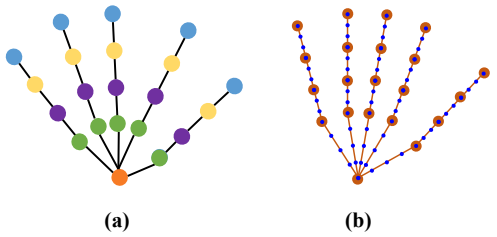

Figure 3. (a) Joint grouping for confidence encoding. Joints with the same color belong to the same group. (b) Brown dots represent joint locations and blue dots denote the sphere centers.

penalizes collisions between the m-th and n-th sphere as follows:

$$L_{coll} = \sum_{m,n} \mathbf{A}_{m,n}^{sphere} max(r_m + r_n - \|c_m - c_n\|_2, 0), \quad (4)$$

where r and c represents the radius and the 3D coordinates of the sphere, respectively; We adopt $\mathbf{A}^{sphere}$ to discard collisions that do not need to be considered, such as the collisions between the palm spheres and the collisions between the spheres in the same bone.

5. Details of Training

The training of our framework includes three stages. In the first stage, the single-view estimation network is trained for 15 epochs with an initial learning rate of 1e-3, which drops to 1e-4 at 10-th epoch. In the second stage, the cross-view fusion network is trained for 10 epochs with an initial learning rate of 1e-3, which drops to 1e-4 at 5-th epoch. In the third stage, the whole network is trained for 10 epochs with an initial learning rate of 1e-4, which drops to 1e-5 at 5-th epoch. Due to the depth offset between the depth image after style transfer and the original depth image, we directly use the original synthetic depth image in the third

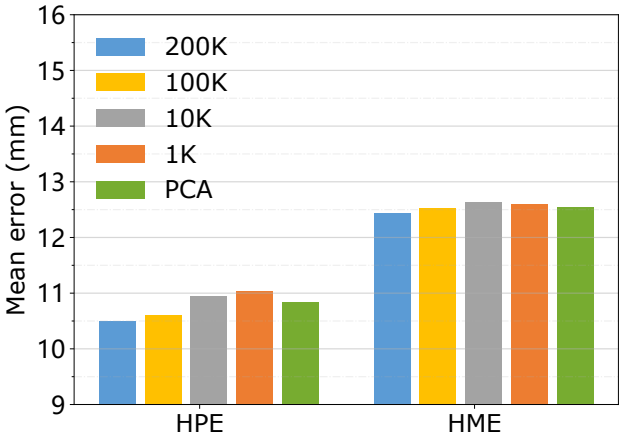

Figure 4. The impact of the amount of synthetic data.

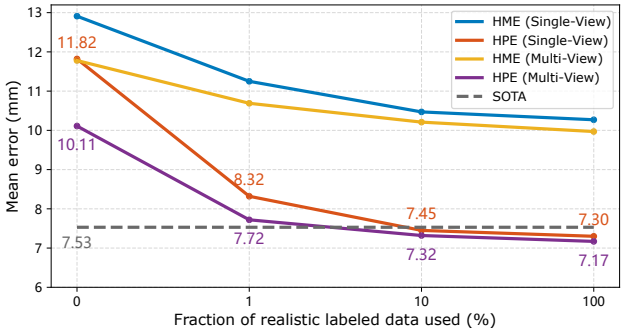

Figure 5. The performances of HPE and HME with different fraction of labeled data on NYU dataset.

stage, which can slightly improve the performance of the network. For synthetic data, we perform an online data augmentation by sampling the hand shape with a normal distribution $N(0; 3)$, the hand scale with a uniform distribution $U(0.8, 1.2)$ and the hand rotation with a uniform distribution $U(0, 2\pi)$. For real data, we perform online data augmentation including random 2D plain rotation ([-180, 180]), random scaling ([0.8, 1.2]) and random translation ([-10, 10]). To generate pseudo multi-view data, we perform data augmentation including random 2D plain rotation ([-180, 180]), random scaling ([0.7, 1.3]), random translation ([-20, 20]) and random erasing. For the constant weights used to balance the self-supervised losses, we set $w_{sm} = 1$, $w_{cm} = 1$, $w_{ms} = 10$, $w_{prior} = 1$, $w_{shape} = 0.1$, and $w_{coll} = 0.01$. For the single-view scenario, when using the 1-stacked network, since the performance of cross-view fusion will decrease, we set $w_{ms} = 1$ to make the self-supervised training more stable. For the trainging of the CycleGAN [14], the whole network is trained using Adam [5] with an initial learning rate of 0.0002. From the 20-th epoch to the 40-th epoch, the learning rate gradually and linearly decreases to 0.

6. Impact of the Training Data

In this section, we investigate the impact of the synthetic data and the amount of labeled data. For synthetic data, we randomly sampled 200K, 100k, 10K and 1K hand pose data from the BigHand2.2M dataset [12] to generate synthetic data respectively. For fairness, we increase the number of epochs of pre-training and fine-tuning as the synthetic data decreases. Since this experiment needs to perform all three training stages and we do not have enough GPUs, we use one-stacked network for this experiment. As shown in Fig. 4, our method is robust to the amount of synthetic data. Even if only adopting 1K synthetic data in the pre-training and fine-tuning, the mean joint error of HPE and HME can still achieve about 11 mm and 12.6 mm. Furthermore, we also try to sample pose data from the PCA prior of MANO. Specifically, we generate the synthetic data by adding random noise (uniform distribution [-3, 3]) to the first ten principal components of the MANO pose space. In this case, compared to the model trained with real pose data, the performance of HPE and HME decreased slightly. The above results demonstrate that our method is robust to the quantity and quality of synthetic data.

In Fig. 5, we show the performance of the two-stacked network trained with different percentages of real labeled data on NYU dataset. By using 100% or even 10% labeled data, our method outperforms SOTA supervised methods, whether in multi-view or single-view scenarios. By using only 1% labeled data, the performance of our method is significantly improved.

7. More Qualitative Results

7.1. NYU Dataset

As shown in Fig. 6, we first compare our method with SOTA strongly supervised methods on some hard samples on the NYU dataset [10]. Specifically, we select three representative strongly supervised methods: 3D CNN based method (V2V) [7], point cloud based method (P2P) [2] and pixel-wise pose regression method (AWR) [4]. In particular, AWR is similar to HPE in our method, but they show significant differences due to different training mechanism. By comparing these hard samples, we draw the following conclusions: (1) For some samples that hand poses are not complex but with relatively extreme views (5425, 8218), our method can better perceive the depth information of the input data and generate a hand pose that better fits with the depth image. (2) Our method can better maintain the hand structure of the predicted pose. For example, for the gesture of making a fist (3657), all strongly supervised methods fail completely. (3) For some complex poses (4273, 5067), the estimation of our method outperforms not only strongly supervised methods but also annotations. (4)

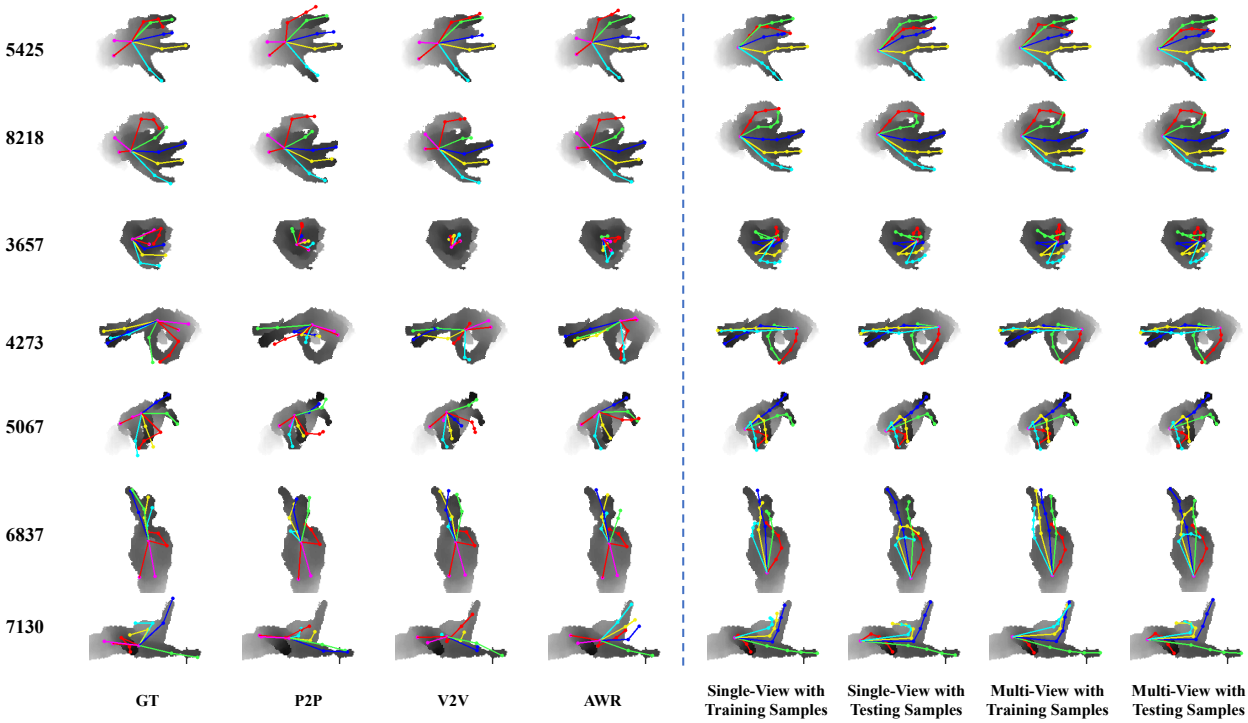

Figure 6. Qualitative results on NYU dataset. Left: the results of some strongly supervised methods. Right: the results of our method with different training data.

For some extremely complex hand poses (6837, 7130), our method can not predict accurate results. However, through self-supervised training on these samples, our method can achieve relatively reasonable prediction.

Then, we further analyze the effect of training data, including the effect of the multi-view data and the effect of training on the testing samples. From Fig. 7, we draw the following conclusions: (1) Adopting multi-view data can significantly reduce the impacts of serious image holes (548), image noise (1777), and self-occlusion (3901, 2895). (2) Our method may be confused by the appearance of some unseen images (6386, 5236), which can be significantly alleviated by performing self-supervised training on these confusing data.

7.2. MSRA Dataset

We give some qualitative results to show that MSRA dataset [9] has serious annotation errors and the overfitting problem of strongly supervised methods. MSRA dataset contains 17 hand gestures for 9 subjects, each with approximately 500 images. We find that almost every gesture has some annotation errors with specific modes, which occur frequently for all subjects. As shown in Fig. 8, the strongly supervised method may overfit these annotation errors and ignore the appearance of the input data. In contrast, our self-supervised method can generate more accurate posture and hand meshes. For the gesture with contact between the fingers (79, 686, 2561, 3484, 3572, 4142, 5011) or the gesture with self-occlusion (4797, 6766), annotations often have serious errors. Even for some simple gestures (1144, 1743, 2198, 5505, 7136), when the hand is rotated globally, the annotations can also be wrong.

References

[1] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2272–2281, 2019. 1

[2] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 475–491, September 2018. 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[4] Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. Awr: Adaptive weighting regression for 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11061–11068, 2020. 3

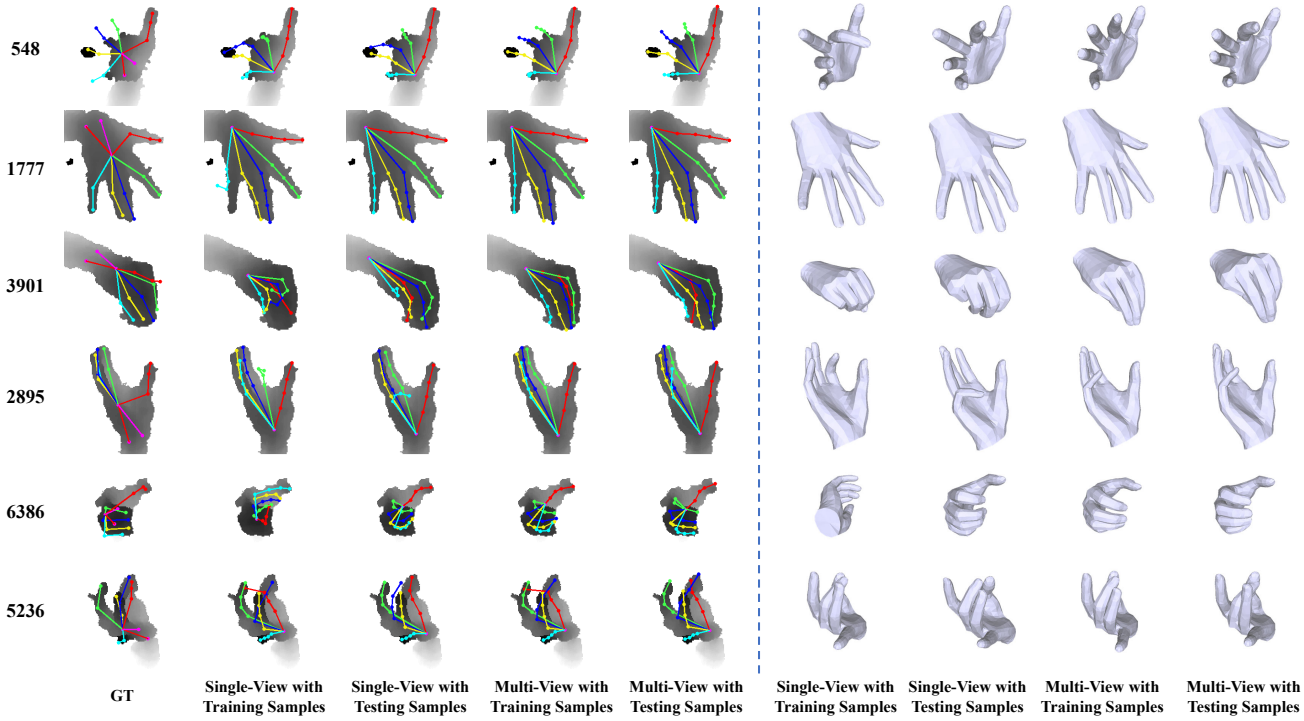

Figure 7. Qualitative results on NYU dataset. Left: the results of HPE with different training data. Right: the results of HME with different training data.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2015. 3

[6] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *European Conference on Computer Vision*, pages 318–334. Springer, 2020. 1

[7] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, June 2018. 3

[8] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 483–499, 2016. 1

[9] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, June 2015. 4

[10] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169:1–169:10, 2014. 3

[11] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16105–16114, 2021. 1

[12] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, July 2017. 3

[13] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019. 1, 2

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3

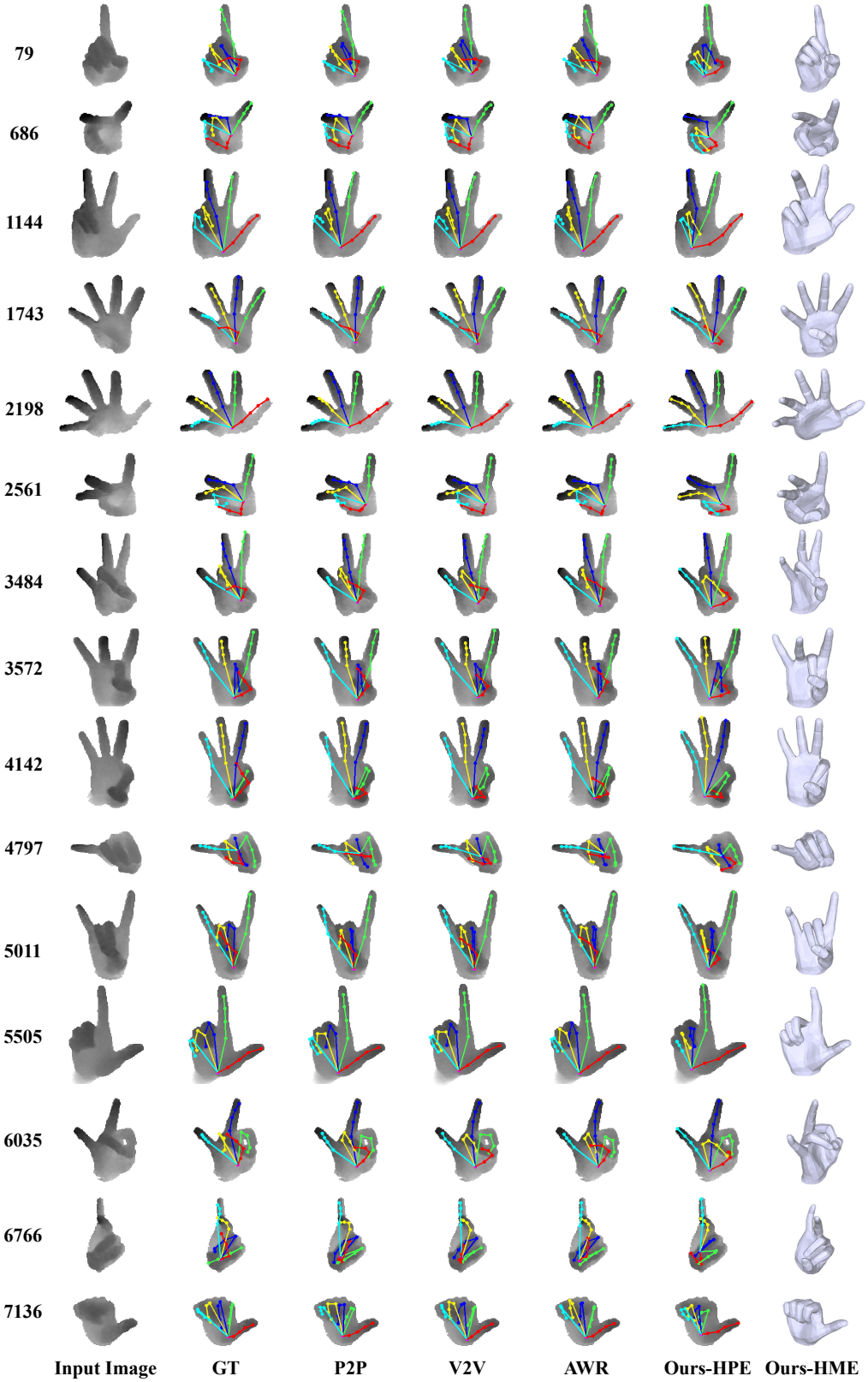

Figure 8. Qualitative results on MSRA dataset. The number represents the frame ID in the test set for MSRA datasets.