Supplementary for Local and Global Point Cloud Reconstruction for 3D Hand Pose and Shape Estimation

Ziwei Yu[1]
yuziwei@u.nus.edu
Linlin Yang[1, 2]
yangll@comp.nus.edu.sg
Shicheng Chen[1]
e0534721@u.nus.edu
Angela Yao[1]
ayao@comp.nus.edu.sg

[1] National University of Singapore, Singapore
[2] University of Bonn, Germany

1 Our Proposed Dataset

1.1 Motivation

Using multi-view cameras is a natural way to obtain a complete point cloud. However, there is to date only one multi-view 3D hand pose dataset– NYU [3]. Furthermore, NYU is primarily used as a depth map dataset; although they also provided the RGB images, these RGB images are of low resolution and quality. FreiHand [6] also records data using a multi-view setup, but the released images are not from corresponding viewpoints. In that sense, it can be regarded only as a single-view dataset containing multiple views rather than a true multi-view dataset.

To fill this gap, we present a new multi-view RGB-D 3D hand pose dataset. We use four RealSense D415 cameras in different views to record 4 RGB-D sequences from 4 subjects and the resolution of our recorded dataset is 640×480. We follow works like [2, 5] and use a 21-joint model to annotate the hand pose. Additionally, we provide hand masks, 2D and 3D joint locations, hand meshes in the form of MANO parameters, real complete hand point clouds and full camera parameters. In particular, we provide extrinsic camera parameters so it is easy for users to use multi-view information. Fig 1 shows a sample image; for more footage, please refer to video 1.

1.2 Annotation

To label our dataset, we develop a mixed semi-supervised system that leveraging the self-supervision technique as proposed in [4]. Initially, we label a small set of images and train a self-supervised network [4] with all unlabeled N frames and synthetic data. We then visualize

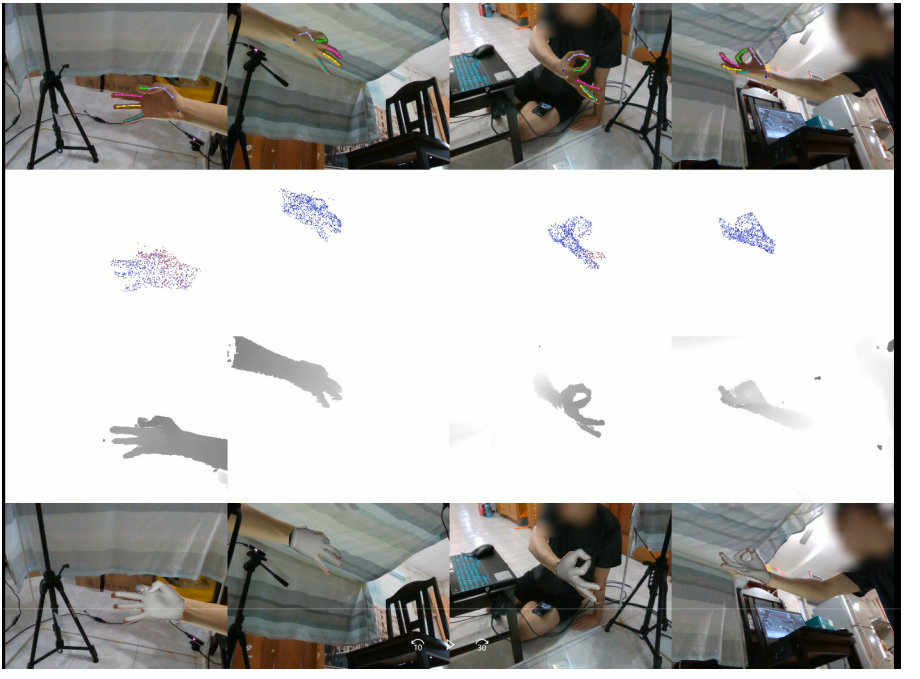

Figure 1: Sequence sample from our proposed MVHand dataset. From top to bottom: RGB images with 2D poses, point cloud ground truth, depth maps and mesh images, respectively. Please go to see our dataset video for more details.

the results using 21 key points and manually label some additional frames with the highest loss before iterating again (see Figure 2). In total, we label 6.5% manually, while 93.5% are automatically labelled via the self-supervision framework of [4]. Our automatically labelled frames have an estimated error of 4.69 mm from the self-supervision, which is in line with the estimated pose error of 5mm for the MANO model [1].

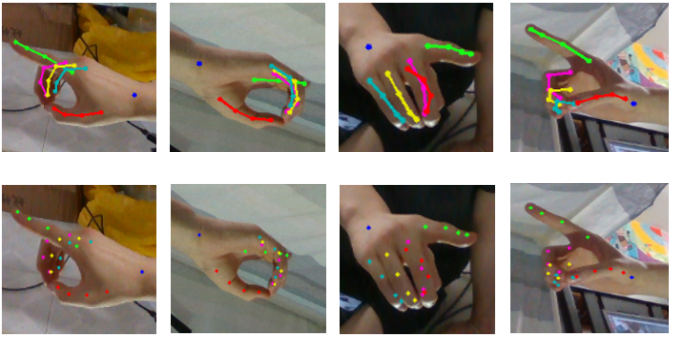

Figure 2: Annotation by using multi-view information

2 Local Partitioning via k-Means

When using real point clouds, the 3D points are unordered, so we cannot directly leverage the template segmentation labels. As an alternative to the k-NN partitioning described in the main paper, we also try using k-means to partition the point clouds randomly (see Figure 3). The local point cloud reconstruction results based on the k-means are similar; with a high enough k, *e.g.*, $k=6$, the partitioning is similar to the manual partition we used for MANO. We show some examples for the local and global point cloud reconstruction results based on the manual MANO partitioning and for K-means with $k=2$ and $k=6$.

(a) Manual (b) Two clusters (c) Six clusters

Figure 3: Local components based on manual partitioning (a) and K-means clustering (b-c).

3 Impact of Initialization

We compare the point cloud reconstruction results for various template initialization in Figs 6-9. The results show that 1) Reconstruction with local templates (k-means, hand labelled) is always better than with global templates. 2) 3D hand initialization is more useful for 3D hand point cloud reconstruction than 2D grid initialization. 3) Our hand-labelled local initialization is better than k-means, verifying the importance of semantic information. This is also verified numerically in Table 1 and 2.

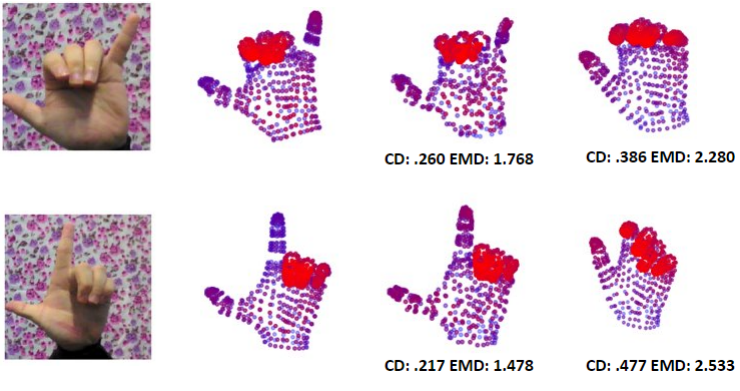

Figure 4: Learning point cloud through our point cloud fitting and MANO layer fitting. The four columns show the RGB input image, point cloud ground truth, our point cloud fitting, MANO layer fitting.

	k-means ($k=2$)	k-means ($k=6$)	3D hand	3D hand (*6)	local 3D hand
STB	0.257	0.251	0.254	0.252	**0.244**
RHD	0.545	0.520	0.550	0.533	**0.450**
Youtube3D	0.292	0.290	0.302	0.295	**0.270**
MVHand	1.112	1.103	1.151	1.105	**0.993**

Table 1: Mean Chamfer Distance per point. 3D hand(*6): increase by six times the number of convolutional channels in the folding layers. The best score is marked in **bold**.

	k-means ($k=2$)	k-means ($k=6$)	3D hand	3D hand (*6)	local 3D hand
STB	1.751	1.680	1.710	1.704	**1.680**
RHD	1.823	1.805	1.840	1.807	**1.781**
Youtube3D	1.220	1.142	1.241	1.182	**0.990**

Table 2: Mean Earth Mover's Distance per point. 3D hand(*6): increase by six times the number of convolutional channels in the folding layers. The best score is marked in **bold**.

4 Point cloud fitting

Fig.4 shows the fitting experiments by using our proposed point cloud method and MANO layer fitting method. We feed the first RGB image as input and the second coloum point cloud as ground truth. As for the MANO fitting, we use the same distance loss to train the MANO layer. Our proposed methods CD and EMD value are smaller than MANO fitting. Through these experiments, compared with our proposed method, fitting a mesh from point cloud is a challenging task.

5 Impact of Model Complexity

The above experiments verifies our local 3D hand initialization's strong performance. To further verify that improvements do not come from an increased model size, we increase by six times the number of convolutional channels in the folding layers (3D hand (*6)) and report the mean CD and EMD for this larger model in Tables 1 and 2. Based on these results, we observe that compared with the standard 3D hand initialization, a larger model (3D hand (*6)) does achieve better results, verifying that increasing model complexity is useful. However, when compared with local 3D hand initialization, our novel local decoder pipeline is still stronger, even though 3D hand (*6) pipeline has nearly the same amount of model parameters, verifying the strengths of our decoder design strategy.

6 Sequence Point Cloud Reconstruction

We show in two videos (STB.mp4 and MVHand.mp4) the point cloud reconstructions over time on MVHand and STB. In the video visualization, the red points are close to the camera and blue points are far. From the videos, we observe that our point cloud reconstruction pipeline can achieve complete, high quality point clouds.

7 Failure Cases

Some failure cases are presented in Fig 5. It can be observed that reconstructions on hand fingers do not achieve good results, compared with other parts of hand. Reasons for failures include object occlusions, two hand interactions, self-occlusions and incomplete inputs. To address these problems, we aim to direct future work on reconstructing hand surfaces under the above situations.

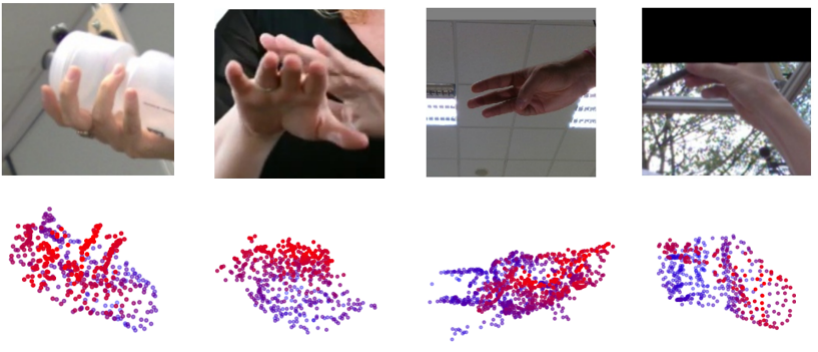

Figure 5: Failure case examples.

References

[1] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6): 245, 2017.

[2] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.

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

[4] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dual grid net: hand mesh vertex regression from single depth maps. In *ECCV*, 2020.

[5] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.

[6] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 813–822, 2019.

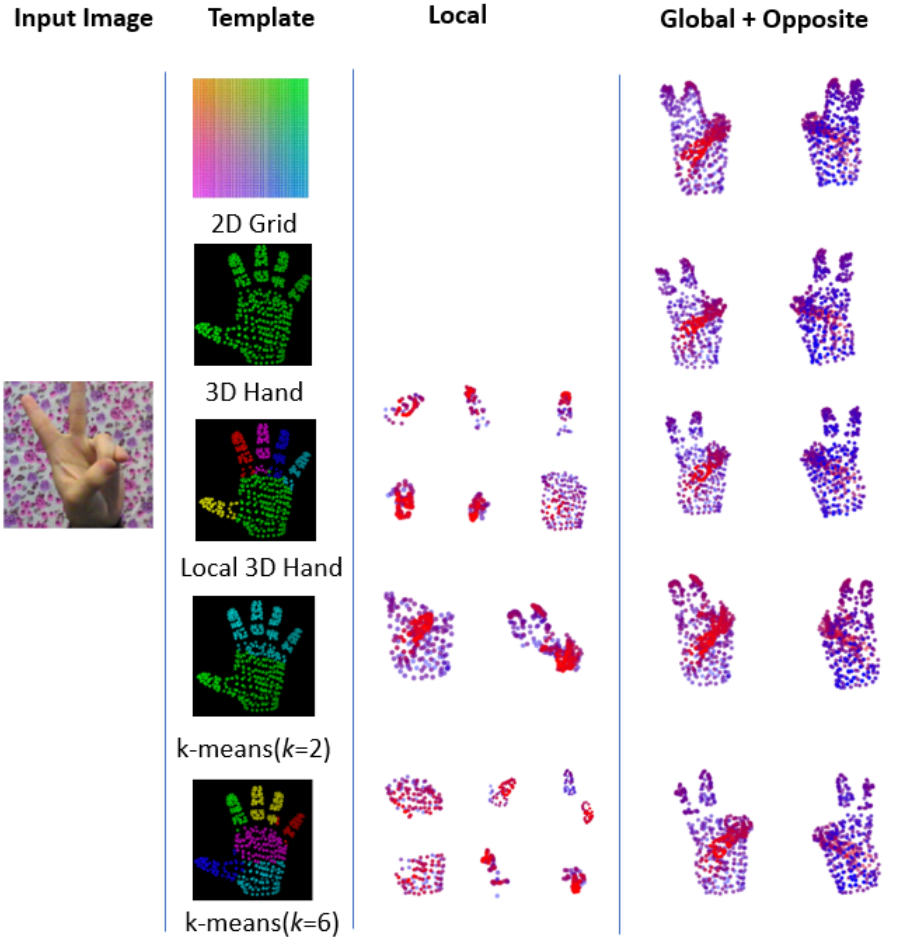

Figure 6: Point cloud reconstruction results based on different templates from STB. The four columns show the RGB input image, templates, local point cloud results, global and opposite camera view's point cloud results.

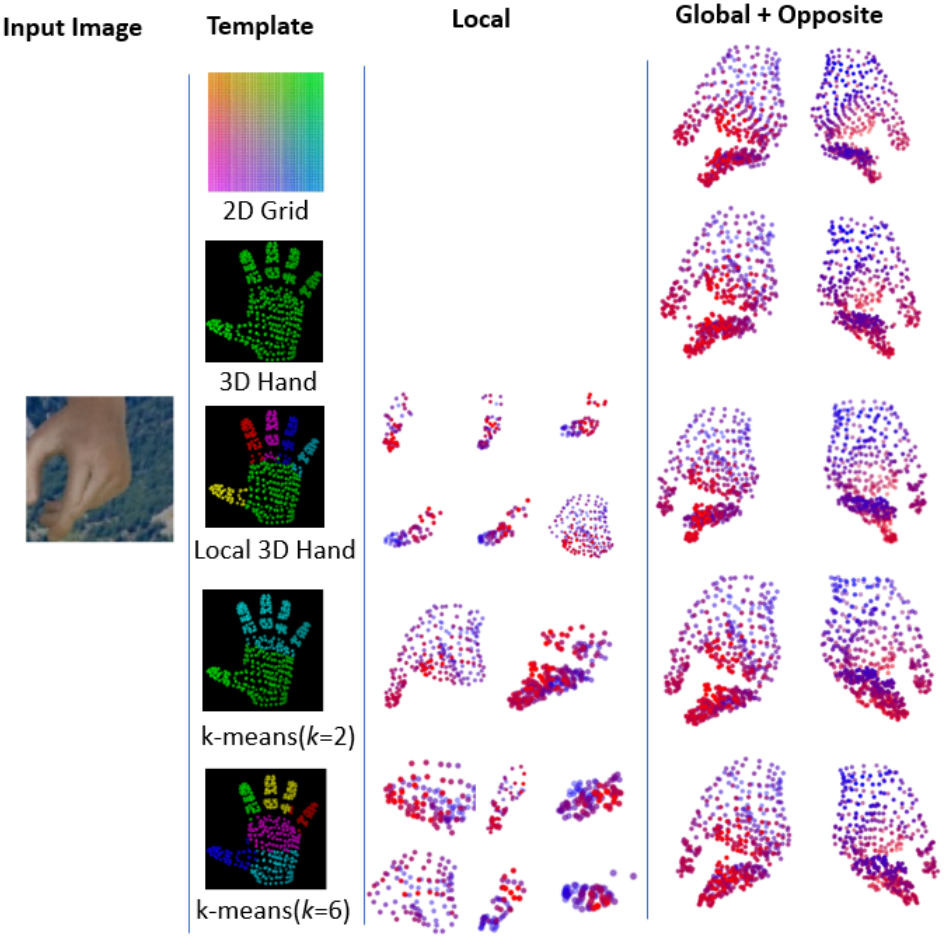

Figure 7: Point cloud reconstruction results based on different templates from RHD. The four columns show the RGB input image, templates, local point cloud results, global and opposite camera view's point cloud results

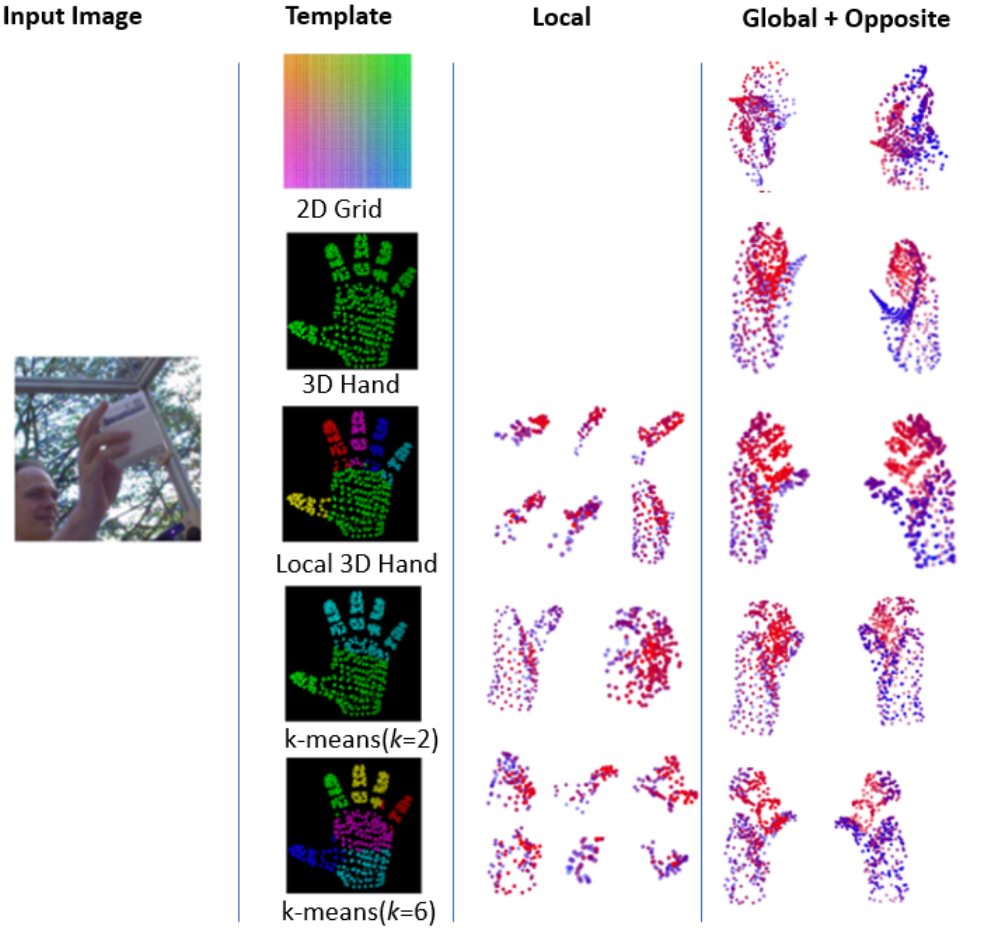

Figure 8: Point cloud reconstruction results based on different templates from FreiHand. The four columns show the RGB input image, templates, local point cloud results, global and opposite camera view's point cloud results.

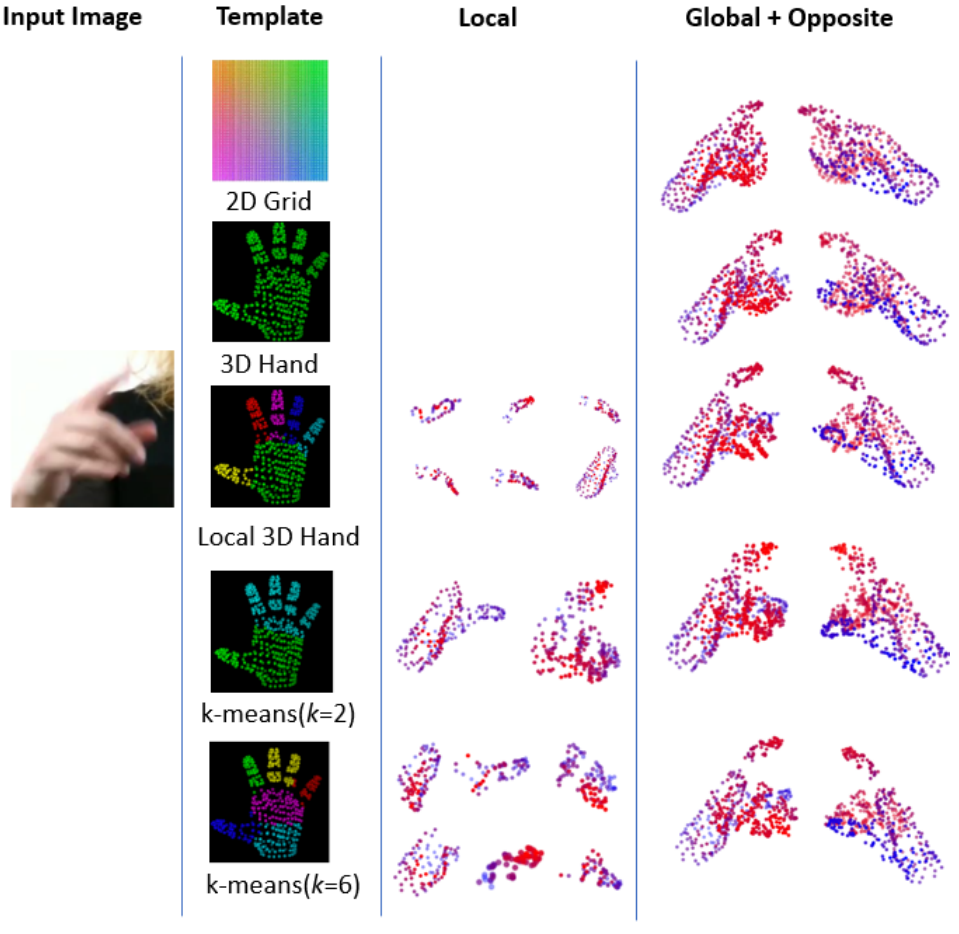

Figure 9: Point cloud reconstruction results based on different templates from Youtube3D. The four columns show the RGB input image, templates, local point cloud results, global and opposite camera view's point cloud results.