Local and Global Point Cloud Reconstruction for 3D Hand Pose 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

Abstract

This paper addresses the 3D point cloud reconstruction and 3D pose estimation of the human hand from a single RGB image. To that end, we present a novel pipeline for local and global point cloud reconstruction using a 3D hand template while learning a latent representation for pose estimation. To demonstrate our method, we introduce a new multi-view hand posture dataset to obtain complete 3D point clouds of the hand in the real world. Experiments on our newly proposed dataset and four public benchmarks demonstrate the model's strengths. Our method outperforms competitors in 3D pose estimation while reconstructing realistic-looking complete 3D hand point clouds.

1 Introduction

The 3D shape and pose of the human hand are critical for augmented and virtual reality applications. To accommodate this form of human-computer interaction, an entire discipline of computer vision is devoted to estimating 3D hand shape and pose. Achieving accurate estimates is extremely challenging due to the hand's high degrees of articulation and self-occlusion. Earlier approaches attempted to combine representations from various viewpoints [7, 8, 9, 10], or transform 2.5D depth maps to 3D representations such as voxels [23, 25], point clouds [21], or meshes [34]. Since 3D voxel models are computationally more expensive than mesh and point cloud models, the latter two are preferable for estimating 3D hand shape and pose.

Current RGB-based methods [2, 19, 46] prefer to estimate hand shape by mapping visual features to the parameters of a parametric model *e.g*. MANO [29]. However, the MANO mesh inherently differs from the real hand surface, resulting in an unnaturally smoothed hand shape. Non-parametric mesh techniques [11, 34] can generate more realistic shapes and account for shape surface details but learning such models requires a substantial amount of annotated mesh data that are non-trivial to collect.

We believe that 3D hand point clouds could serve as a non-parametric alternative to meshes. Unlike meshes, point clouds are unordered and do not require predefined topological structure for surface. Point clouds are easily obtained from various sources, *e.g.* depth cameras, laser scanners, and other 3D representations. Moreover, a 3D hand point cloud's density is easily adjustable; depending on the resolution requirements, we can down- or up-sample the number of points from the surface of the hand. The use of point clouds for 3D hand pose and shape estimation is limited [9, 10, 21, 37]. The work most closely related to ours is [37], which estimates a point cloud from different modalities, including RGB images. Their estimated point cloud, is only of the camera-facing surface. A complete point cloud would provide more complete geometric information, as shown previously in 3D body pose estimation [1, 39, 45].

This paper proposes a unique point cloud reconstruction technique for determining the full hand shape and pose from RGB images. We put forth a combined local and global representation for learning a detailed 3D latent representation for the hand. To reconstruct an accurate and high-resolution point cloud of the hand, we draw inspiration from point cloud architectures like FoldingNet [38] and AtlasNet [12]. However, our work is novel in that we design a new template initialization specifically for 3D hand recovery. Our template is flexible and enables us to pre-distribute the 3D points in a configuration more useful for reconstructing the 3D hand. Additionally, we offer a semantic grouping strategy for reconstructing the local and global point clouds that correspond to the individual fingers.

Existing RGB-based 3D hand pose datasets do not have any corresponding (complete) 3D point cloud data. As such, we sample from the surface of the MANO mesh model to generate point clouds for existing datasets. Additionally, we introduce a new multi-view RGB-D dataset and illustrate the usefulness of our methodology on real-world point clouds recovered from depth images to validate our methodology on real-world data.

Our contributions are summarized as follows:

- We propose a unique framework for 3D point cloud reconstruction of the hand with a customized 3D hand template. To our knowledge, our system is the first to reconstruct a complete 3D hand point cloud rather than just the camera-facing surface.
- We propose an effective combined local and global point cloud reconstruction method which captures more detailing than a single global model.
- We introduce a multi-view RGB-D hand pose dataset with 3D joint annotations, fitted MANO parameters and depth-map based 3D point clouds.
- Evaluation on four public benchmarks and our own newly proposed dataset verifies that our framework can outperform state-of-the-art approaches for 3D hand pose estimation while being able to reconstruct high-quality point clouds.

2 Related Works

Point Cloud Reconstruction methods in deep learning primarily learn unordered representations by examining the intrinsic 3D structure. Tree-based models [6, 18, 43] represent point clouds through *k*-d trees. Other works propose innovative network designs, including PointNet [28], PointCNN [22] and RNN-based models [40]. To date, most of these approaches [18, 28, 38, 40] concentrate on point cloud reconstruction from 3D Lidar images. Our purpose, in comparison, is to recreate point clouds from RGB images.

3D Hand pose estimation often use depth maps or RGB pictures as input. The 2.5D-depth map is fed into deep neural networks [27] to obtain heatmaps which are then lifted to a 3D-hand joint location [7, 8, 9, 10, 25, 30, 35]. Recent papers attempt to estimate the hand pose of an RGB image along with additional modalities (*e.g.*, depth or mask) as weak labels [3, 37] or as intermediate forms of supervision [15, 46, 47]. Others [8, 10, 21] have tried to convert depth maps to obtain an incomplete point cloud of the camera-facing viewpoint to help predict 3D pose. To the best of our knowledge, we are the first to integrate point cloud reconstruction with 3D pose estimation from monocular RGB images.

3D Hand shape estimation typically in the form of meshes, can be more challenging than pose estimation because it needs to simultaneously predict the mesh topology and vertex locations. Most approaches [16, 17, 19, 26, 46] reconstruct the mesh by leveraging the parametric MANO model [29]. Using MANO allows these works to directly regress shape (and pose) parameters, which sit in a more tractable and lower-dimensional space. For RGB inputs, the standard approach [19, 46] is to firstly estimate 2D joint locations and then iteratively regress the MANO pose and shape parameters. Non-parametric methods leverage either a fully convolutional network [34] or a graph convolutional network [11] to directly regress mesh vertices. Unlike these above approaches, we aim to recover complete point clouds, which we believe to be a more flexible 3D representation than a mesh.

3 Methodology

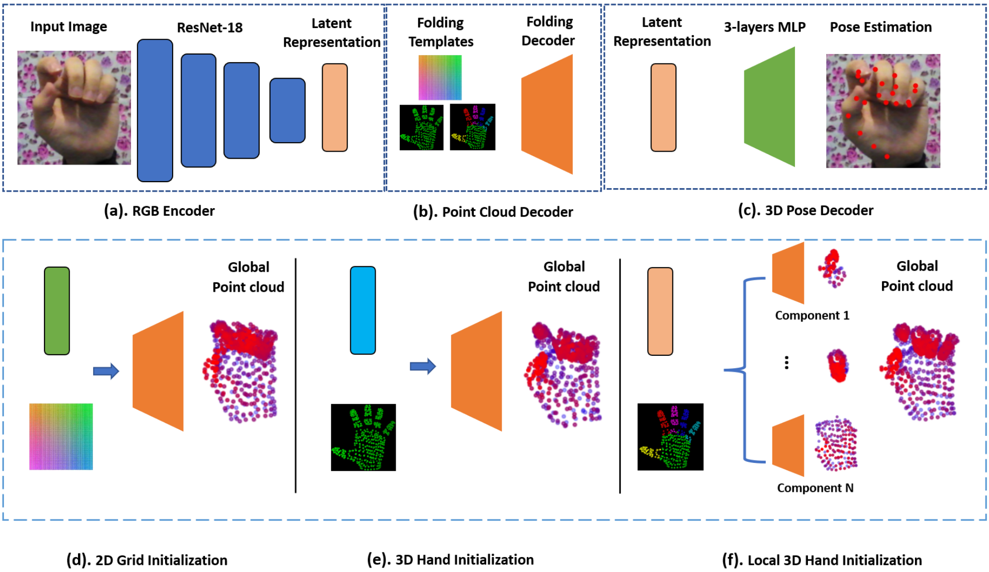
3.1 RGB Encoder and 3D Pose Decoder

Our framework has an RGB image encoder, a point cloud decoder, and a 3D pose decoder (see Fig. 1). The encoder converts the image into a latent representation. Our core contribution is in learning this latent representation such that it is rich and expressive for accurate 3D pose estimation (Sec. 3.1) and complete 3D point cloud reconstruction (Sec. 3.2).

The image encoder encodes a single 256×256 RGB image $\mathbf{x}$ into a latent representation $\mathbf{z} \in R^{512}$. The pose decoder converts $\mathbf{z}$ into the hand pose $\mathbf{J} \in \mathbb{R}^{3 \times 21}$, *i.e.* the 3D coordinates of 21 hand joints. We use the same backbone models as other encoder-decoder frameworks [19, 36, 37]. For the encoder, we fine-tune a ResNet-18 backbone and use the final layer output fed into a fully connected layer as the latent representation $\mathbf{z}$. For the 3D pose decoder, we use a three-layer fully connected multi-layer perceptron (MLP) with 128 hidden units per layer. To train the pose decoder, we use an L2 loss, *i.e.*, $\mathcal{L}_{Pose} = ||\mathbf{J}_{pred} - \mathbf{J}_{gt}||_2$, where the $\mathbf{J}_{gt}$ and $\mathbf{J}_{pred}$ denote the ground truth and estimated hand pose, respectively.

3.2 3D Point Cloud Reconstruction

To reconstruct a 3D hand point cloud $\hat{\mathcal{S}} \in \mathcal{R}^{N \times 3}$ from $\mathbf{z}$, we follow the decoding architecture of FoldingNet [38]. FoldingNet's decoder is a series of MLPs that deform or "fold" a template set of points into their final 3D positions by conditioning on the encoded latent representation. Assuming that we are given an RGB image and an accompanying ground truth point cloud $\mathcal{S}$, the encoder-decoder pair can be learned via the Chamfer distance (**CD**) and Earth Mover's distance (**EMD**). Both distances are computed between $\mathcal{S}$ and estimated point cloud set $\hat{\mathcal{S}}$ ($|\mathcal{S}|$ = $|\hat{\mathcal{S}}|$). If $s \in R^{3\times1}$ and $\hat{s} \in R^{3\times1}$ are ground truth and predicted 3D



(a). RGB Encoder (b). Point Cloud Decoder (c). 3D Pose Decoder

(d). 2D Grid Initialization (e). 3D Hand Initialization (f). Local 3D Hand Initialization

Figure 1: **Overview of our point cloud reconstruction and pose estimation pipeline**.Our proposed has an RGB image encoder(a), a point cloud decoder (b) and a 3D pose decoder (c). We experiment with three template initializations in (d) to (f) and observe that the most effective is to use local 3D initializations that represent the various semantic components of the hand (f).

points respectively, then the Chamfer distance d_{CD} is defined as

$$d_{CD}(\mathcal{S},\hat{\mathcal{S}}) = \frac{1}{|S|}\sum_{s\in S}\min_{\hat{s}\in\hat{S}}||s-\hat{s}||_2^2 + \frac{1}{|\hat{S}|}\sum_{\hat{s}\in\hat{S}}\min_{s\in S}||s-\hat{s}||_2^2, \qquad (1)$$

where the first term is the average distance of all predicted points to the closest ground truth point and the second term is the average distance of all ground truth points to the closest predicted point. The Earth-Mover's distance d_{EMD} factors in the point-to-point assignment problem. Let $\phi : \hat{S} \to S$ be a bijection, *i.e.* for all $s \in S$, there is a uniquely matched point $s \in \hat{S}$. The optimal bijection is unique and invariant over the above point sets.

$$d_{\text{EMD}}(\hat{\mathcal{S}},\mathcal{S}) = \min_{\phi:\hat{S}\to S}\sum_{s\in\hat{S}}||s-\phi(s)||_2. \qquad (2)$$

Initialization Templates: The original FoldingNet initializes the template point set on a 2D lattice grid (see Fig. 1(d)). The decoder then "folds" these points into a 3D surface structure. Using a 2D lattice grid is well-suited for class- or 3D-shape-agnostic reconstruction since it makes no prior assumptions. However, we posit that it is non-ideal and inefficient for 3D hand shape estimation. It is more direct to initialize the point set to follow some canonical 3D hand. Therefore, we propose a 3D hand template initialization (see Fig. 1(e)). By fixing the initial spatial distribution to follow a hand, we simplify the learning of the decoder as it reduces the extent of folding required. The comparison of different initialization can be found in Supplementary Sec. 3.

Local Reconstruction: We further modify the FoldingNet decoder to do reconstruction locally. In particular, we are interested in high-fidelity reconstructions of the fingers because they contain much of the pose information. It is therefore intuitive to offer separate representations to each of the fingers. To that end, we assign a local decoder to the palm and each of the fingers (see Fig. 1(f)) and apply the Chamfer and Earth Mover's distance to these local point cloud sets. We leverage the (ordered) MANO vertices to separate the point cloud into individual components and their associated ground truths (see Sec. 3.3). To ensure that the components fit together, we also apply the two distances globally across the complete point set to arrive at the loss in Eq. 3.

Similar to other point cloud reconstruction methods [5, 20, 24, 37], we use the above two distance to learn the point cloud reconstruction. Specifically, we apply the distances in a global sense, *i.e.* d^G to the point cloud set of the entire hand, as well as in a local manner, *i.e.* d^L to a subset of points that correspond to the local components. The final loss on the point cloud $\mathcal{L}_{\text{pc}}$ is a sum of these distances, *i.e.*

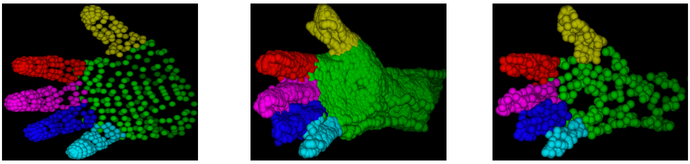
$$\mathcal{L}_{\text{pc}} = \sum_i \left(d_{CD}^{L_i} + d_{EMD}^{L_i} \right) + d_{CD}^{G} + d_{EMD}^{G}, \tag{3}$$

where i indexes the local components. While we can introduce weighting hyperparameters to the terms in Eq. 3, we keep them equally weighted out of simplicity and as what have been done in [37].

3.3 Generating Ground Truth Point Clouds

Existing RGB-based hand pose benchmarks have ground truth 3D poses but no point cloud information. As an alternative, we leverage the MANO [29] model and sample from the fitted mesh surface to obtain a set of 3D points. More specifically, MANO parameterizes a triangular mesh $\mathcal{M} \in R^{N \times 3}$ with parameters $\{\vec{\beta}, \vec{\theta}\}$, where $\vec{\beta} \in R^{10}$ signify the shape parameters and $\vec{\theta} \in R^{K \times 3}$ are pose parameters. Similar to [19], we fit the MANO model $\mathbf{J}_{\text{MANO}} \in R^{21 \times 3}$ to the ground truth 3D poses $\mathbf{J}_{\text{gt}} \in R^{21 \times 3}$ by minimizing the following L2 objective: $\min_{\vec{\beta}, \vec{\theta}} ||\mathbf{J}_{\text{gt}} - \mathbf{J}_{\text{MANO}}||_2$. Based on the fitted mesh vertices, we upsample and then randomly downsample to obtain a set of 3D points distributed on the hand surface.

Local Component Assignment: A local reconstruction requires assigning each template point to a fixed local component, *i.e.* which points in the ground truth should be used to evaluate each local reconstruction? For the MANO-generated point clouds, we manually split the 778 vertices into six semantic portions corresponding to the palm, thumb, and four fingers (see Fig. 2(a)). Real point clouds, however, are unordered. To assign points, we firstly estimate the 3D pose and the MANO parameters. We then apply a simple k-nearest neighbor classifier (k= 3) to match each point to a MANO vertex and give it the same component as the MANO vertices' partition. We can also re-sample as necessary (see Fig. 2 (b) and (c)).



(a) MANO-generated (b) Depth map-based (c) Depth downsampled

Figure 2: Segmentation Transfer from the MANO vertices to real depth point clouds.

	STB	RHD	FreiHand	YouTube3D	MVHand
Modality	real rgbd	syn. rgbd	real rgb	real rgb	real rgbd
Resolution	640 × 480	320 × 320	240 × 240	mixed	640 × 480
Hands	single	two	single	single	single
Subjects	1	20	32	–	4
Viewpoints	2	1	1	1	4
MANO params	✗	✗	✓	✓	✓
Frames	36K	44K	36K	51K	83K

Table 1: Comparison of proposed MVHand Dataset with other RGB benchmarks.

3.4 MVHand Dataset

Methods of generating point clouds based on MANO are straightforward approaches to add point cloud data to the current RGB datasets. However, they are not fully representational, as the MANO mesh is a smoothed approximation of a genuine hand's surface. Real-world point clouds from the multi-view depth maps are dense and noisy; variations between the two may be observed in Fig. 2. To verify that our framework also works on real-world point clouds, we need a multi-view RGB-D dataset, *i.e.* RGB images as input and multi-view depth for constructing ground truth point clouds. The only such dataset to date is the NYU Hand Pose Dataset [33]. However, this dataset does not provide camera extrinsics, so it is not possible to composite the views into complete point clouds. Furthermore, the recordings were done by Kinect V1 sensors and the depth maps are very noisy.

To cover this gap, we record a new multi-view RGB-D dataset, which we call MVHand. Similar to the BigHand [41] and Ho3D [13] datasets, we record our dataset with Inter RealSense D415 cameras. We use four cameras, each at a range of approximately 50cm from the hand. This is within the manufacturer's recommended range of 45cm to 2m, which is then specified to have a depth error of $< 2\%$[1]. At this distance we observe that manually labelled 2D keypoints in one view (see Fig. 3) can accurately project onto the other views.

To obtain the point clouds, we firstly segment the hand in the depth image via thresholding, and then project the four views into a complete hand. The final complete 3D point cloud ground truth is obtained by sampling from these depth maps. Specifically, we remove any isolated outlier points via filtering. Additionally, we downsample the points in overlapping areas from the different views to ensure that the points in the point cloud are evenly distributed. Fig. 4 shows some sample point clouds from our dataset; our point clouds well-represent the original hand shape and are close to fitted MANO mesh vertices. The mean per-point Chamfer distance from point cloud to mesh vertex 7.88mm, with a standard deviation of 0.72 mm. We direct the reader to the Supplementary for further details.

The 3D hand poses are annotated using the same 21-joint hand model as [32, 47]. We use a semi-automated method, combining human annotations (around 7% of frames) with the self-supervision method of [34] (see Supplementary 1.2 for details). The automatically labelled frames have an estimated average joint error of 4.69mm when evaluated against manually labelled samples, which is close to the manual label errors of Megahand [14].

Table 1 compares the statistics of MVHand with various hand benchmarks. MVHand provides RGB and depth maps from the four views, 3D joint positions, fitted MANO parameters, hand masks, and all camera intrinsics and extrinsics. Fig. 3 shows sample frames; more visualizations can be found in the Supplementary.

[1] https://www.intelrealsense.com/depth-camera-d415/

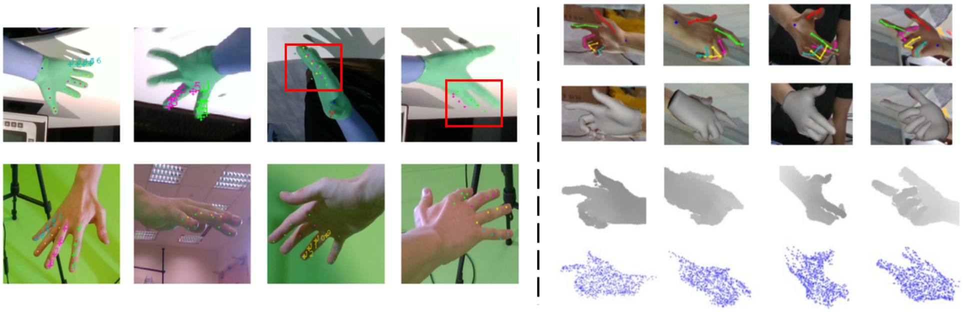


Figure 3: Left: Different distances for annotation settings. Top row shows that recording with a larger distance between hand and camera. Bottom row shows that distance is smaller than 50 cm. Our manual annotations are based on the first column image and project into other three views. Right: Samples from our proposed MVHand dataset. We highlight the error region with red box.

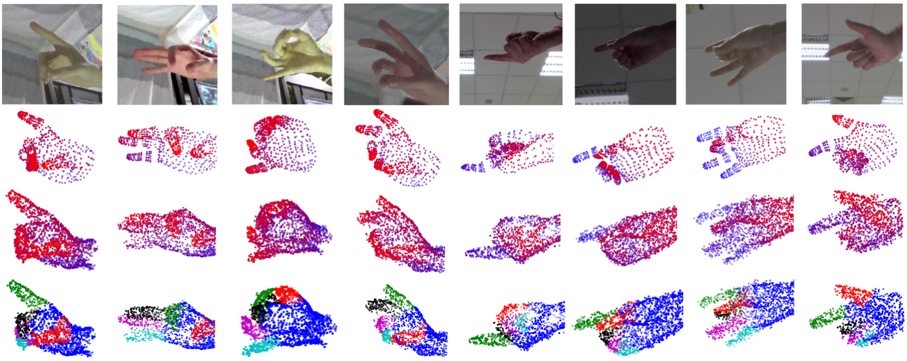


Figure 4: MVHand dataset point clouds visualization. These four rows show RGB images, mesh vertices, our complete point clouds, our point clouds segmentation by using K-Nearest Neighbors algorithm based on mesh vertices.

4 Experiments

Datasets and Metrics: We test on four standard RGB-based hand post estimation benchmarks in addition to our own recorded MVhand dataset. **RHD** [47] is a synthetic dataset of rendered hands with 42k training images and 2.7k testing images. **STB** [44] features videos of a single person's left hand in front of 6 real-world indoor backgrounds. We use the 15k/3k training/test split proposed in [47]. **FreiHAND** [48] is a challenging multi-view RGB dataset of hand-object interactions. **YouTube3D** Hands-in-the-Wild [19] features images curated from Youtube videos with a 47k/1.5k/1.5k image training/validation/test split.

To evaluate 3D pose accuracy, we use area under the curve (AUC) on the percentage of the correct keypoint (PCK) score, where PCK is calculated using various error thresholds [47]. We also evaluate the mean 3D joint distance (mm) to ground truth according to mean-per-joint-position-error (MPJPE). To evaluate the reconstructed point clouds, we compute the mean Chamfer and Earth-Mover's distances as per Eq 1 and 2.

Implementation Details: We optimize using ADAM to train the point cloud reconstruction firstly and then for 3D pose estimation. For RGB to point cloud encoder-decoder, we use

Method	RHD	STB
Zimm.'17 [47]	30.42	8.68
Spurr'18 [31]	19.73	8.56
Yang'19a [36]	19.95	8.56
Bouk.'19 [2]	16.78*	9.76
Yang'19b [37]	**13.14**	7.05
Iqbal'18 [15]	13.82	8.01*
Ours(w/o rec.)	15.80	7.72
Ours(full)	13.38	**6.71**

Method	FreiHand	YouTube3D	MVHand
Zimm.'19 [48]	11.0	–	–
Bouk.'19 [2]	23.43*	19.24*	
Chen'21 [42]	11.8	–	–
Choi'20 [4]	**7.6**	–	–
Yang'19b [37]	12.35*	18.76*	15.12*
Iqbal'18 [15]	13.52*	19.32*	15.27*
Ours(w/o rec.)	13.90	22.50	17.80
Ours(full)	9.60	**18.50**	**14.50**

Table 2: Comparison of MPJPE (mm) with SOTA. "w/o rec." means without using our point cloud reconstruction pipeline, otherwise, "full". The best score is marked in **bold**. * indicates results based on released source code of [2] and [37], and our re-implementation of [15].

an initial learning rate of 0.001, a weight decay of 1e-6 and a batch size of 32. Afterwards, we fine-tune the RGB encoder while learning the 3D pose decoder, using an initial learning rate of 0.0001 and weight decay of 1e-6.

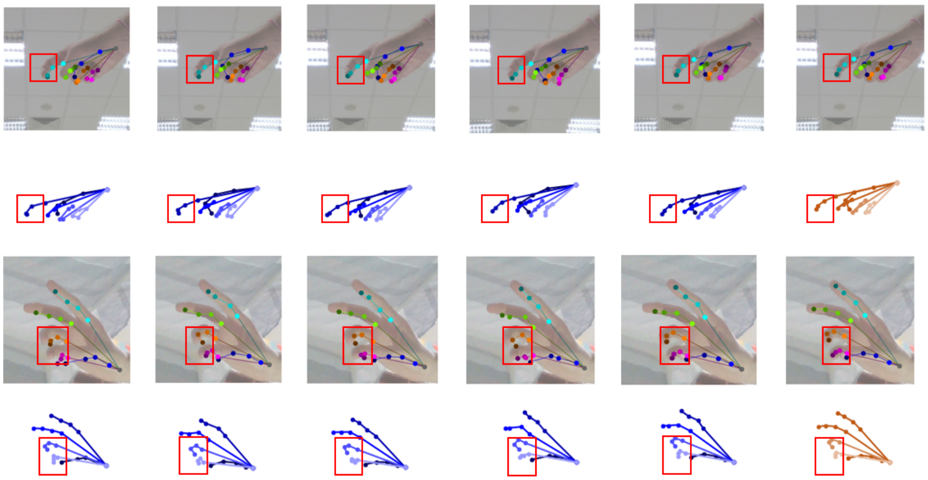
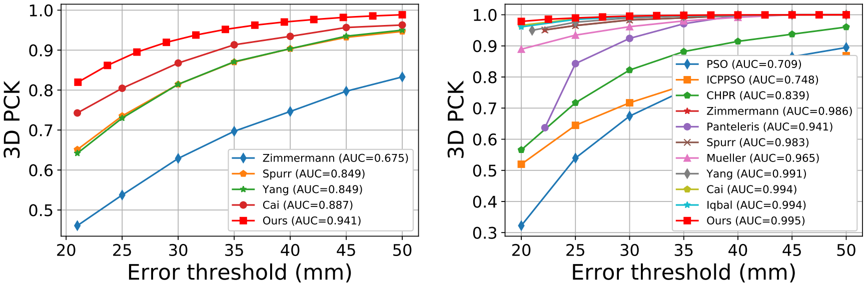


Figure 5: 2D and 3D pose visualization. Left to right column: Bouk.'19 [2], Iqbal'18 [15], Yang'19b [37], Ours(w/o rec.), Ours(full), ground truth. We highlight the differences among predictions and the ground-truth poses with red boxes.

4.1 Comparison of Pose Estimates

Table 2 compares our 3D pose estimation accuracy. On RHD, our proposed framework surpasses most other approaches [15, 31, 36, 47]. Our MPJPE is comparable to Yang *et al.* [37] despite their use of perspective correction and additional modalities like 2D heatmaps. On STB, we have the lowest MPJPE. Figure 6 (a) and (b) compare the 3D PCK curve on RHD and STB respectively. On both datasets, our proposed method obtains the highest AUC. There are very few published results on Freihand and YouTube3D. For FreiHand, our MPJPE (9.6mm) is lower than [2, 48]. Choi'20 [4] reports 7.6mm, however, they use 2D poses from other pre-trained models as input. YouTube3D does not provide the hand scale, so we use 40mm as reference bone length [2] to evaluate their test set. Table 2 also compares pose esti-

[2] Reference bone length as defined by Freihand; 40mm comes from the STB dataset



(a) RHD AUC

(b) STB AUC

Figure 6: Comparisons with state-of-the-art methods on RHD [47] and STB [44] .

	Chamfer Distance			Earth Mover's Distance		
	2D grid	3D hand	local 3D	2D grid	3D hand	local 3D
STB	0.26	0.25	**0.24**	1.75	1.71	**1.68**
RHD	0.59	0.55	**0.45**	2.01	1.84	**1.78**
YouTube3D	0.31	0.30	**0.27**	1.32	1.24	**0.99**
MVHand	1.20	1.15	**0.99**	–	–	–

Table 3: Mean CD and EMD per point; the best score is marked in **bold**.

mation accuracy on our proposed MVHand dataset. As a baseline, we follow [15] to directly regress the 3D hand pose with the 2.5D pose representation using a hold-one-subject out test split. Our full model's MPJPE, at 14.5mm, surpasses this baseline by 0.77mm. More qualitative results are shown in Fig. 5.

4.2 Ablation Studies

Point Cloud Decoder: We remove the point cloud decoder and directly learn an image-to-pose encoder-decoder with the same architecture components as our current model. Table 2 shows that the setting (w/o rec.) results in a higher error than the full model with the point-cloud decoder. This is the case for all benchmark datasets and our MVHand; on average, the error is 20% higher than the full model. These results verify that the point cloud reconstruction helps to learn a better latent representation for 3D pose estimation.

	Chamfer Distance				Earth Mover's Distance			
	[2]	Ours (3D)	[37](Sur.)	Ours (Sur.)	[2]	Ours (3D)	[37](Sur.)	Ours (Sur.)
STB	.367	**.243**	.146	**.113**	2.34	**1.68**	**3.136**	5.294
RHD	.627	**.450**	**.195**	.299	**1.95**	1.78	**4.434**	5.134

Table 4: Mean CD and EMD per point; "Sur." indicates CD and EMD on a 2.5D space as per [15], since [37] estimates only the camera-facing surface of the hand. The best score is marked in **bold**. Surface result values are scaled by 1000.

Template Initialization: We compare the point cloud reconstructions from the 2D grid, 3D hand, and local 3D hand initialization in Table 3 using CD and EMD to evaluate; on both distances, a smaller value is better. We omit EMD on our MVHand dataset as the number of points (1038) is too large to compute within a feasible time. The results in Table 3 support the strength of our local 3D hand initialization.

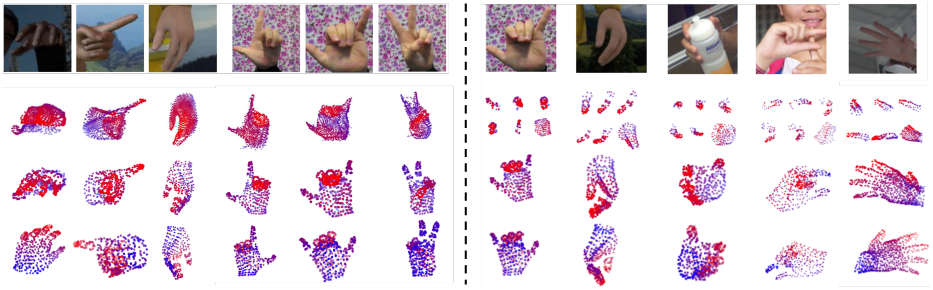


Figure 7: Left: Reconstruction results compared with [37] for RHD and STB. Top to bottom: RGB images, incomplete reconstructions from [37], our complete reconstruction from the same viewpoint and opposite viewpoint. Right: Top to bottom: original RGB images from five datasets, the component-wise point cloud reconstructions from our method, aggregated into "Global" and its opposing view. Red points are close to the camera, and blue are far.

4.3 Point Cloud Reconstructions

We visualize sample reconstructions in Figure 7 for RHD and STB. Figure 7 also compares to [37]'s method, which reconstructs the camera-facing surface point cloud. In Figure 7, we visualize samples from five datasets and observe that our reconstruction results are not only complete but also of higher quality, especially in distinguishing the individual fingers. Furthermore, we also visualize our component-wise point cloud reconstruction results and these results verify the effectiveness of our local and global framework in reconstructing high fidelity point clouds.

As there are no other works that make complete point cloud reconstructions, we cannot make any direct quantitative comparisons. Instead, we make two indirect comparisons in Table 4. First, we compare with [2], which directly regresses MANO parameters. We project the estimated MANO parameters into a 3D point cloud in the same way as described in Sec. 3.3. We find that our 3D results are in both CD and EMD for the STB dataset and better in CD for the RHD dataset. We also compare with [37], though as their method only recovers surface poitn clouds, we project our complete point cloud into 2.5D space as per [15]. Although our EMD results are worse than [37], we believe the projection process accumulates some errors. When visualized, however, our point clouds are much cleaner and of higher quality than [37] (see Fig. 7).

5 Conclusion

This paper proposed a framework for reconstructing a complete 3D point cloud from RGB images. In learning an RGB-point cloud encoder-decoder, we also learned a rich latent representation that can be decoded into an accurate 3D hand pose. To improve the quality of the point clouds, we introduced two template initializations. To verify our method on real-world hand point cloud data, we introduced MVHand, a new multi-view RGB-D dataset. Experimental results showed that our proposed method achieves comparable or better performance than existing 3D hand pose and shape estimation methods. In future work, we will explore the use of point clouds to resolve self-occlusions of the hand.

6 Acknowledgments

This research is supported by Singapore Ministry of Education (MOE) Academic Research Fund Tier 1 T1251RES1819.

References

[1] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2293–2303, 2019.

[2] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[3] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[4] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pages 769–787. Springer, 2020.

[5] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[6] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[7] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3593–3601, 2016.

[8] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1991–2000, 2017.

[9] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018.

[10] 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 (ECCV)*, pages 475–491, 2018.

[11] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10833–10842, 2019.

[12] T Groueix, M Fisher, VG Kim, BC Russell, and M Aubry. Atlasnet: a papier-mâché approach to learning 3d surface generation (2018). *arXiv preprint arXiv:1802.05384*, 11.

[13] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3196–3206, 2020.

[14] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (TOG)*, 39(4):87–1, 2020.

[15] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018.

[16] David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5610–5619, 2016.

[17] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2540–2548, 2015.

[18] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017.

[19] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4990–5000, 2020.

[20] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020.

[21] Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11927–11936, 2019.

[22] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018.

[23] Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7113–7122, 2020.

[24] Priyanka Mandikal, Navaneet KL, and R Venkatesh Babu. 3d-psrnet: Part segmented 3d point cloud reconstruction from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[25] 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, 2018.

[26] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

[27] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[29] 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.

[30] Ayan Sinha, Chiho Choi, and Karthik Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4150–4158, 2016.

[31] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018.

[32] 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.

[33] 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.

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

[35] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 793–802, 2019.

[36] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9877–9886, 2019.

[37] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2335–2343, 2019.

[38] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud autoencoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

[39] Pengfei Yao, Zheng Fang, Fan Wu, Yao Feng, and Jiwei Li. Densebody: Directly regressing dense 3d human pose and shape from a single color image. *arXiv preprint arXiv:1903.10153*, 2019.

[40] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 403–417, 2018.

[41] 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, 2017.

[42] Chen Yujin, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10451–10460, 2021.

[43] Wei Zeng and Theo Gevers. 3dcontextnet: Kd tree guided hierarchical learning of point clouds using local and global contextual cues. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[44] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016.

[45] Tianshu Zhang, Buzhen Huang, and Yangang Wang. Object-occluded human shape and pose estimation from a single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2020.

[46] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2354–2364, 2019.

[47] 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.

[48] 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.