Efficient Virtual View Selection for 3D Hand Pose Estimation

Jian Cheng[1,2*], **Yanguang Wan**[1,2*], **Dexin Zuo**[1,2], **Cuixia Ma**[1], **Jian Gu**[3],
Ping Tan[3,4], **Hongan Wang**[1], **Xiaoming Deng**[1†], **Yinda Zhang**[5†]

[1]Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences   [3]Alibaba   [4]Simon Fraser University   [5]Google

Abstract

3D hand pose estimation from single depth is a fundamental problem in computer vision, and has wide applications. However, the existing methods still can not achieve satisfactory hand pose estimation results due to view variation and occlusion of human hand. In this paper, we propose a new virtual view selection and fusion module for 3D hand pose estimation from single depth. We propose to automatically select multiple virtual viewpoints for pose estimation and fuse the results of all and find this empirically delivers accurate and robust pose estimation. In order to select most effective virtual views for pose fusion, we evaluate the virtual views based on the confidence of virtual views using a light-weight network via network distillation. Experiments on three main benchmark datasets including NYU, ICVL and Hands2019 demonstrate that our method outperforms the state-of-the-arts on NYU and ICVL, and achieves very competitive performance on Hands2019-Task1, and our proposed virtual view selection and fusion module is both effective for 3D hand pose estimation. The code is available in project webpage https://github.com/iscas3dv/handpose-virtualview.
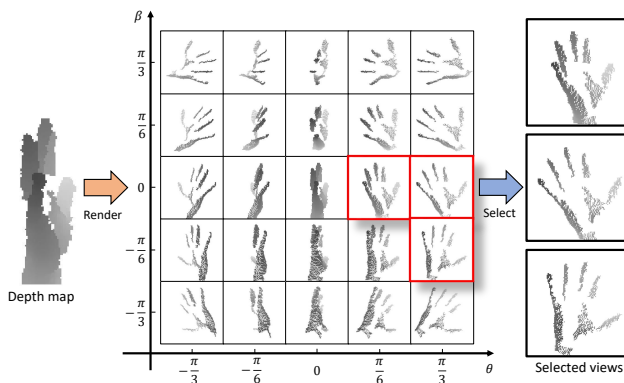
Figure 1: Illustration of view selection for 3D hand pose estimation. The view of the original depth image may not be suitable for pose estimation. We select suitable views for pose estimation from the uniformly sampled virtual views. θ and β represent the azimuth angle and the elevation angle of candidate virtual views, respectively.

1. Introduction

Hand pose estimation plays a key role in many applications to support human computer interaction, such as autonomous driving, AR/VR, and robotics (Erol et al. 2007). Given an input image, the goal of hand pose estimation is to estimate the location of hand skeleton joints. While many works take color images as input, methods built upon depth images usually exhibit superior performance (Sun et al. 2015). Prior arts often use depth image and 2D CNN networks to regress 3D hand joints, or apply point-net based models (Qi et al. 2017; Ge et al. 2018) on point clouds converted from the depth using camera intrinsic parameters. Although great progress on hand pose estimation from depth has been made in the past decade, the existing methods still can not achieve satisfactory hand pose estimation results due to the severe viewpoint variations and occlusions caused by articulated hand pose.

To address the occlusion and viewpoint variation challenges, existing methods often rely on data alignment that transforms the input depth input into a canonical space. However, this process is either done in 2D space (Sun et al. 2015; Ye, Yuan, and Kim 2016) that do not fully respect the 3D nature of the depth image, or in hand-crafted canonical space, e.g. via PCA (Ge et al. 2018) or axis-aligned bounding box (Ge et al. 2016), that are not engaged in a joint optimization with the full model and thus the performance may not be at its best. In contrast, learning based data alignment is more optimal as demonstrated in many previous work (Jaderberg et al. 2015), and back to the domain of pose estimation, this is mostly achieved in automated best viewpoint selection in multiple camera system (Pirinen, Gärtner, and Sminchisescu 2019; Gärtner, Pirinen, and Sminchisescu 2020) via reinforcement learning. However, in the scenario when only one single depth is available, reinforcement learning does not typically perform well due to the limited inputs, and there is limited research study if viewpoint selection is necessary and possible with a single input depth.

We claim that viewpoint selection is very important for 3D hand pose estimation even from just a single depth. As a perspective projection of the 3D geometry, a depth image

---

*These authors contributed equally.
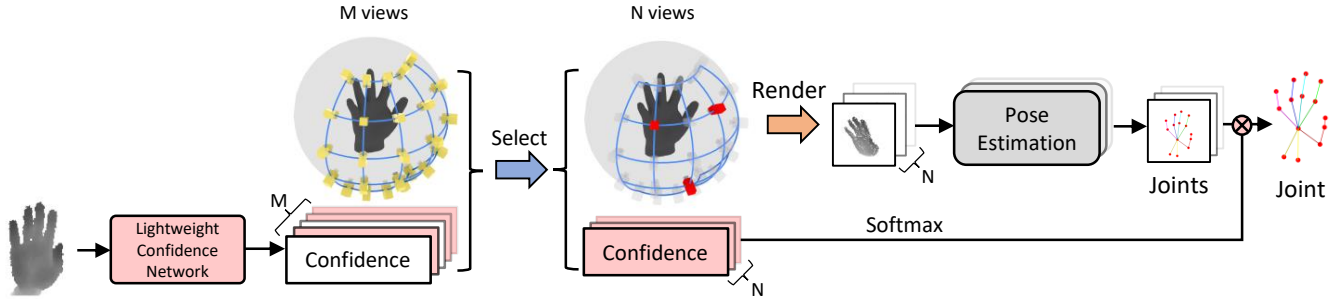
†indicates corresponding author.

Figure 2: Illustration of our virtual view selection and fusion pipeline for 3D hand pose estimation.

can be projected into the 3D space as a point cloud and rendered into depth maps from another viewpoints for 3D hand pose estimation. On one hand, the same amount of error on 2D perspective views may correspond to dramatically different errors in 3D space. One the other hand, machine learning models may favor some typical viewpoints than the others (See Fig. 1 for an example). Therefore, finding the proper virtual viewpoint to re-project the given single input depth could be critical to further improve hand pose estimation, which has been well-studied in terms of architecture.

In this paper, we propose a novel virtual view selection and fusion module for 3D hand pose estimation methods from single depth, which can be easily integrated into the existing hand pose estimation models to enhance the performance. Instead of selecting just one best viewpoint, we propose to automatically select virtual viewpoints for pose estimation and fuse the results of all views, and find this empirically delivers accurate and robust pose estimation. To achieve this, we re-render the input depth into all candidate viewpoints, and train a viewpoint selection network to evaluate the confidence of each virtual viewpoint for pose estimation. We find this empirically works well but slows down the run-time speed when the viewpoint candidate pool is large. To alleviate the computation issue, we adopt network distillation and show that it is possible to predict the view confidence without explicitly re-project the input depth.

The contributions of our method can be summarized as follows: We propose a novel deep learning network to predict 3D hand pose estimation from single depth, which renders the point cloud of the input depth to virtual multi-view, and get 3D hand pose by fusing the 3D pose of each view. We then show that the view selection can be done efficiently without sacrificing run-time via network distillation. Extensive experiments on hand pose benchmarks demonstrate that our method achieves the state-of-the-art performance.

2. Related Work

The 3D hand pose estimation methods take depth images as input and estimate the locations of hand skeleton joints. Prior arts include (Moon, Chang, and Lee 2018; Ge, Ren, and Yuan 2018; Rad, Oberweger, and Lepetit 2018; Wan et al. 2018; Xiong et al. 2019). (Ge, Ren, and Yuan 2018), (Deng et al. 2020) and (Moon, Chang, and Lee 2018) use 3D representation of depth to estimate the hand pose, and

(Wan et al. 2018), (Rad, Oberweger, and Lepetit 2018) and (Xiong et al. 2019) use 2D representation of depth to get the hand pose. Anchor-to-Joint Regression Network (A2J) (Xiong et al. 2019) can predict accurate pose with efficiency, which sets up dense anchor points on the image and obtains the final joint locations by weighted joint voting of all anchor points. Although impressive results are obtained with these prior arts, these networks perform worse under occlusion or severe viewpoint variation.

The most related work to us is Ge et al. (Ge et al. 2016), which shares the same thoughts with us that the input viewpoint may not be ideal and projects the input depth into the front, side and top view for pose estimation. However, the number of the selected virtual views is fixed (i.e. 3), and the view selection strategy is hand-crafted but not trained in an end-to-end manner. Different to them, we proposed a learnable virtual view selection and fusion module for 3D hand pose estimation, which can adaptively select informative virtual viewpoints for point cloud rendering, and fuse the estimated 3D poses of these views.

In the realm of pose estimation, viewpoint selection is also achieved in multiple camera system using reinforcement learning (Pirinen, Gärtner, and Sminchisescu 2019; Gärtner, Pirinen, and Sminchisescu 2020). These works use reinforcement learning methods to select a view sequence suitable for 3D pose estimation. However, they all require a multi-view capture setup, and cannot be used for a single input depth image. Moreover, these methods are time-consuming, because views are selected in sequence, and thus reduce the inference efficiency.

3. Approach

3.1 Overview

In this section, we introduce our virtual view selection and fusion approach for 3D hand pose estimation from a single depth image (Fig. 2). We first convert a depth image into 3D point clouds, and uniformly set up candidate virtual views on the spherical surface centered in the hand point clouds. The point cloud is then rendered into candidate views as depth maps, which are then fed into a network to predict the confidence. A 3D pose estimation network then predicts the 3D

hand pose from view with top-N confidence [1], and finally fuses the pose with regard to the confidence to achieve accurate 3D hand pose results. To reduce the computational cost, we also design an efficient lightweight network by model distillation to predict the view confidence from the input depth itself, which saves the computation cost of point cloud rendering if the pool of candidate view is large.

3.2 Virtual Multi-View Hand Pose Estimation

We first explain the idea of virtual multi-view hand pose estimation. Inspired by Ge et al. (Ge et al. 2016) that the original camera view may not be optimal for hand pose estimation, we hypothesize that denser virtual view sampling should be more beneficial and propose to exploit rendered depth on multiple virtual views to estimate 3D hand pose (Fig. 3).
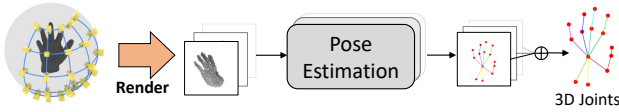

Figure 3: Our virtual multi-view hand pose estimation baseline.

Candidate Virtual Views   We first define a set of virtual camera views to re-render the input depth map. Specifically, we first convert the depth map of the hand into point clouds, and uniformly sample a series of virtual cameras on the spherical surface centered in the hand point clouds. Note that large camera rotation may cause severe occlusion issue for depth rendering, and thus we only keep the cameras close to the input camera view. In practice, we keep 25 virtual cameras uniformly sampled from the zenith angle in $[-\pi/3, \pi/3]$ and the azimuth angle in $[-\pi/3, \pi/3]$ on the sphere. Refer to Fig. 8 in Sec. 4.3 for illustration.

Virtual View Depth Map Rendering   The point cloud from the input depth image can now be re-projected to each of the virtual cameras. Note that since the depth images does not provide a complete hand shape, the rendered depth may be partially incomplete or contain wrong occlusion. However, we found empirically that these does not confuse the pose estimation network when the virtual camera is not too far from the input. Therefore, we did not complete the rendered image as the face rendering methods (Zhou et al. 2020; Fu et al. 2021). We also implemented a parallel rendering process using CUDA to speed up the rendering process.

3D Hand Pose Estimation from Single Depth   For each rendered depth image, we use the Anchor-to-Joint regression network (A2J) (Xiong et al. 2019) as the backbone for 3D hand pose estimation for its great run-time efficiency and competitive accuracy. In practice, any other 3D hand pose estimation models from depth can be taken as the backbone.

We use the loss function from A2J and briefly explain here for self-contain. The loss function L_{A2J} of A2J consists of the objective loss L_{obj} and the informative anchor

---

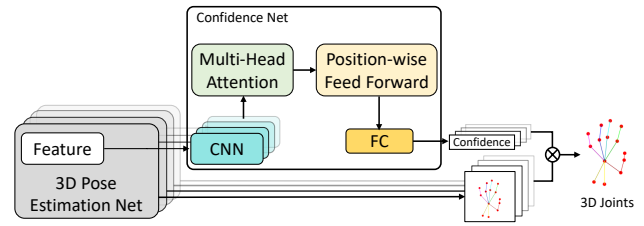[1]N is a hyper-parameter that controls the run-time efficiency and pose accuracy.

Figure 4: Confidence network. The network uses CNN to extract features from the intermediate features of pose estimation net, and then uses multi-head attention to fuse the multi-views visual features. We use FC and Softmax to map the fused multi-view features to confidence of each view.

point surrounding loss L_{info}. The objective loss L_{obj} aims to minimizing the error between the predicted hand joints with the anchor points and the ground truth hand joints, and the informative anchor point surrounding loss L_{info} aims to selecting informative anchor points located around the joints to enhance the generalization ability. The loss function L_{A2J} can be formulated as follows

$$L_{A2J} = \lambda L_{obj} + L_{info} \tag{1}$$

where λ is the loss weight to balance L_{obj} and L_{info}, and it is set to $\lambda = 3$.

Multi-View Pose Fusion   In the end, we run a fusion stage to combine the predicted 3D hand poses from virtual camera views. In the very basic version, we transform the 3D joints of each view to the original camera coordinate system with the camera extrinsic parameters, and get the final hand pose prediction by averaging the transformed hand poses.

3.3 Virtual View Selection

The method proposed in the previous section indeed significantly improves the hand pose estimation accuracy, however suffers from run-time efficiency issue. Essentially, the virtual multi-view baseline runs single view pose estimation network multiple times, thus needs linearly proportional computation power w.r.t the size of virtual camera pool size (e.g. 25 in our case). In another perspective, the pose prediction on some virtual views may not be great, e.g. due to sparsity or wrong occlusion, and may further hurt the overall performance when fused with an average. Therefore, we propose a view selection algorithm that can choose a small number of high impacting camera views without the necessity of running all of them into the pose estimation network.

Confidence Prediction   The key of our view selection algorithm is a confidence network that evaluates the importance of each candidate virtual view (Fig. 4). To save the computation cost, the confidence network takes the high-level feature from A2J as input. It then extracts features using convolutional neural network, and then uses the multi-head attention in (Vaswani et al. 2017) to fuse multi-view features. The multi-head attention mechanism can direct the network to focus on the views which play more important role, e.g. provide more accurate pose or complementary information for fusion. Finally, we use a fully connected layer
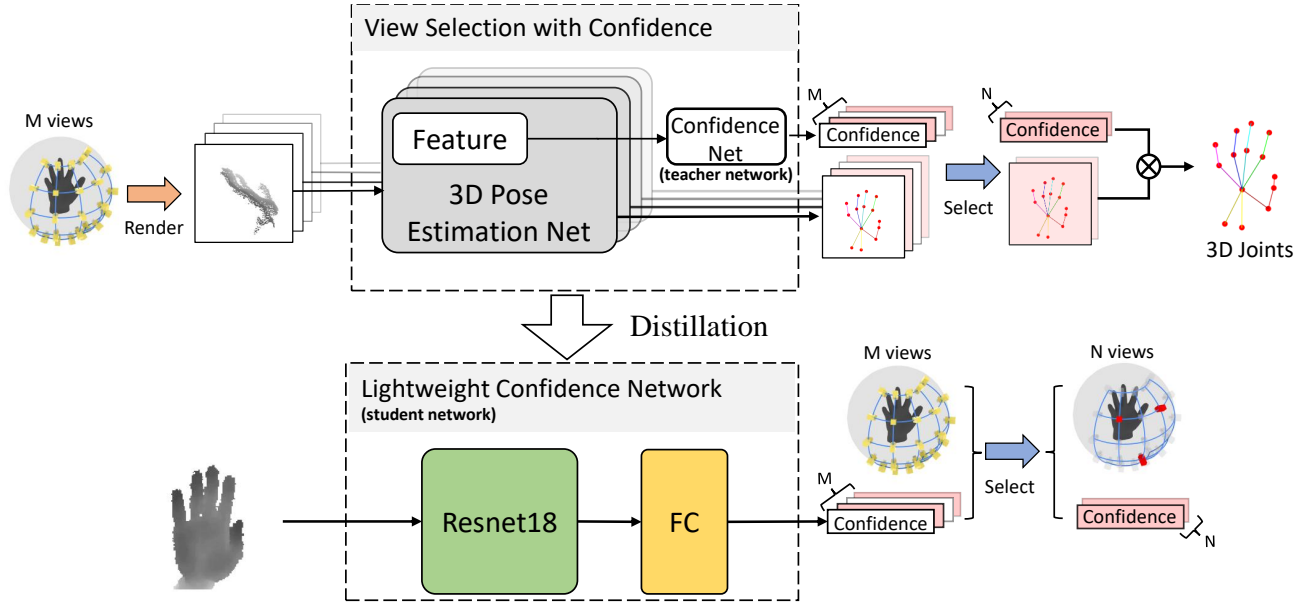
Figure 5: View selection with confidence. We use multi-view depth map data to train a "teacher" confidence network for view selection based on confidence (top), and we train a "student" lightweight confidence network for efficient view selection through network distillation (bottom).

to map the feature of each view to a scalar, i.e. the confidence of each view.

Virtual View Selection and Fusion With the predicted confidence for each virtual camera viewpoint, we can perform view selection by picking the views with top-N confidence from all M candidate views ($N < M$). Empirically, larger N tends to produce more accurate pose (See Sec. 4.3) but needs more computation. In practice, we found that picking $N = 3$ in $M = 25$ virtual views can already beat most of the SoTA methods.

With the pose estimated from selected views, we can fuse hand poses and get final output $\hat{\mathbf{J}}$ as:

$$\hat{\mathbf{J}} = \sum_{i=1}^{N} c_i (\mathbf{R}_i \hat{\mathbf{J}}_i + \mathbf{t}_i) \qquad (2)$$

where $\hat{\mathbf{J}}_i$ is the predicted hand joints of the i-th selected view, c_i is the confidence of the i-th selected view after softmax, and $[\mathbf{R}_i, \mathbf{t}_i]$ is the known camera extrinsic parameters of the i-th selected view.

Joint Training During the training stage, we do not give direct supervision to the confidence of each view since they are simply unknown. Instead, we jointly train confidence network with the pose estimation network, which is supervised only on the final pose accuracy. The joint loss L_J is formulated as:

$$L_J = \sum_{i=1}^{K} L_\tau (\|\mathbf{J}_i - \hat{\mathbf{J}}_i\|) \qquad (3)$$

where $\hat{\mathbf{J}}_i$ is the estimated i-th joint by multi-view pose fusion (See Eq. (2)), $\mathbf{J}_i$ is the ground truth location of the i-

th joint, K is the number of hand joint, and $L_\tau(\cdot)$ is the smooth$_{L1}$ loss function.

Note that the behavior of confidence network might be different with varying N since the predicted confidences have been used at both selection and fusion stage. For the selection stage, only the ranking of the confidence matters; and for the fusion stage, the precise value of confidences of the chosen views also matters. Therefore, for ideal performance, the confidence network should be trained for each specific N. In practice, however, a single model trained with $N = M$ can still work for most of the varying N, with slightly worse but still reasonable performance.

Distillation for Efficient Confidence Prediction With view selection, we are able to reduce the number of forward pass of the pose estimation network from the total number of virtual views M to a much smaller number of selected views N. This reduces enormous computation, but we still observe noticeable frame per second (FPS) drops mostly due to two reasons: 1) The input depth still needs to be rendered to all candidate camera views for confidence estimation. 2) The confidence network still needs to process all the rendered depth, though designed to be light-weighted but still costly to run the multi-head attention.

We resort to network distillation to alleviate this run-time issue. Specifically, we take our confidence network as the teacher network, and train an even more light-weighted student network (a ResNet-18 followed by a fully connected layer). More importantly, the student network takes only the original depth as the input and directly outputs confidence for all $M = 25$ views. This effectively removes the necessity of re-rendering the input depth to all virtual cameras.

To train the student network, we directly take the confi-

dence predicted by teacher network as the ground truth. The loss is defined as

$$L_{light} = \sum_{i=1}^{M} L_\tau(\beta(c_i - \hat{c}_i)) \qquad (4)$$

where β is the scaling factor set to 100, $\hat{c}_i$ is the confidence ground truth from teacher network, i.e. the multi-head attention network from M views, and c_i is the student network output.

Once the training is done, the teacher network is no longer needed during the inference, and student network can support the view selection and fusion in an efficient way.

3.4 Implementation Details
We train and evaluate our models on a workstation with two Intel Xeon Silver 4210R, 512GB of RAM and an Nvidia RTX3090 GPU. Our models are implemented within Py-Torch. Adam optimizer is used; the initial learning rate is set to 0.001 and is decayed by 0.9 per epoch. In order to conduct data augmentation, we randomly scale the cropped depth map, jitter the centroid of the point cloud, and randomly rotate the camera when rendering multi-view depth. For all smooth$_{L1}$ loss, the switch point between quadratic and linear is set to 1.0.

Our network consists of the 3D pose estimation network (i.e. A2J), the teacher confidence network and the lightweight student confidence network. The network input is 176×176 hand region cropped from the input depth, and we use ResNet-50 as the backbone of A2J. We first train the 3D pose estimation network and the teacher confidence network together, the loss can be formulated as:

$$L_{viewsel} = L_{A2J} + \gamma L_J \qquad (5)$$

where $\gamma = 0.1$ is the factor to balance the loss terms.

Then we fix the parameters of the two networks and train the lightweight student confidence network with L_{light}.

4. Experiments
4.1 Datasets and Evaluation Metric
NYU Hand Pose Dataset (NYU) (Tompson et al. 2014) contains 72,757 frames for training and 8,252 frames for testing. 36 hand joints are annotated, but we use only a subset of 14 hand joints for evaluations following the same evaluation protocol in (Tompson et al. 2014).

ICVL Hand Pose Dataset (ICVL) (Tang et al. 2014) contains 331,006 frames for training and 1,596 frames for testing. 16 hand joints are annotated.

Task 1 of Hands19 Challenge Dataset (Hands19-Task1) (Armagan et al. 2020) contains 175,951 training depth images from 5 subjects and 124,999 testing depth images from 10 subjects, in which 5 subjects overlap with the training set. This dataset is very challenging because of its exhaustive coverage of viewpoints and articulations.

Evaluation Metric We evaluate the hand pose estimation performance using standard metrics proposed in (Tang et al. 2014), i.e. mean joint error and the percentage of test examples that have all predicted joints within a given maximum distance from the ground truth.

4.2 Comparison with State-of-the-art Methods

Methods	NYU	ICVL
HandPointNet (Ge et al. 2018)	10.54	6.93
DenseReg (Wan et al. 2018)	10.21	7.24
P2P (Ge, Ren, and Yuan 2018)	9.05	6.33
A2J (Xiong et al. 2019)	8.61	6.46
V2V (Moon, Chang, and Lee 2018)	8.42	6.28
AWR (Huang et al. 2020)	7.37	5.98
Ours-1view	7.34	5.16
Ours-3views	6.82	4.86
Ours-9views	6.53	4.77
Ours-15views	6.41	**4.76**
Ours-25views	**6.40**	4.79

Table 1: Comparison mean joint 3D error (mm) and ranking result with state-of-art methods on NYU dataset and ICVL dataset. "Ours-1view", "Ours-3views", "Ours-9views" and "Ours-15views" are the results of our method with selected 1, 3, 9 and 15 views from 25 uniformly sampled views, respectively. "Outs-25views" denotes the results of our method with 25 uniformly sampled views.
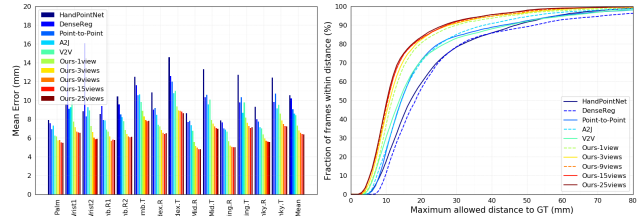

Figure 6: Comparison of our proposed method with state-of-the-art methods on NYU dataset. Left: mean joint error per hand joint. Right: the percentage of success frames over different error thresholds.

In this experiment, we use our "student" lightweight confidence network for view selection and pose fusion.

We first compare our method with the state-of-the-art methods on NYU dataset and ICVL dataset. DenseReg (Wan et al. 2018), A2J (Xiong et al. 2019) and AWR (Huang et al. 2020) directly use the depth map for pose estimation. Hand-PointNet (Ge et al. 2018), P2P (Ge, Ren, and Yuan 2018) and V2V (Moon, Chang, and Lee 2018) use the 3D representation of the depth map for pose estimation.

Table 1 shows the mean joint error. Fig. 6 shows the percentage of success frames over different error thresholds and the error of each joint. We do not show AWR in Fig. 6 because the best prediction results on NYU dataset are not released by AWR. Note that with just 1 view selected, we already outperform all the other methods. And when more view selected, the performance keeps improving. This clearly indicates that our view selection is effective in finding a better virtual view for pose estimation, and more views are benefical through the confidence based fusion.
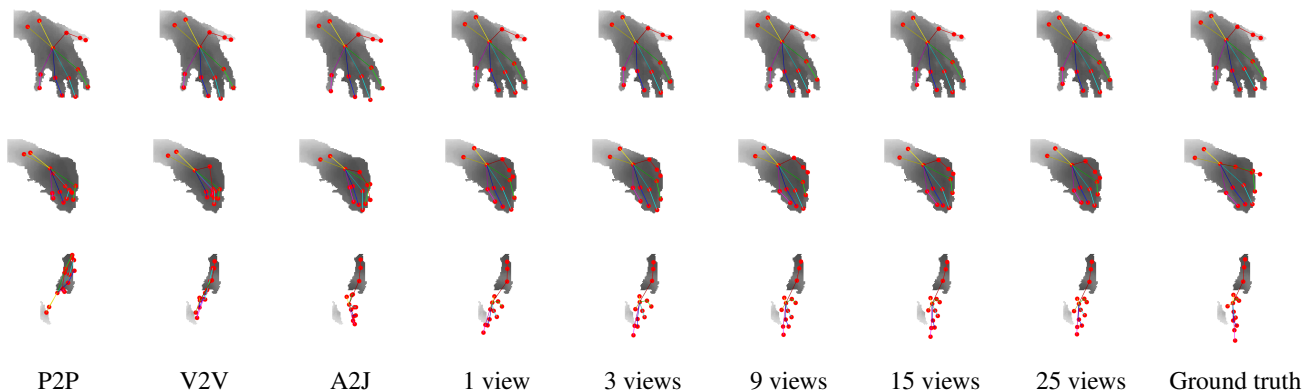
P2P	V2V	A2J	1 view	3 views	9 views	15 views	25 views	Ground truth

Figure 7: Comparison visualization results with state-of-art methods on NYU dataset. "1 view", "3 views", "9 views" and "15 views" are the results of our method with selected 1, 3, 9 and 15 views from 25 uniformly sampled views, respectively. "25 views" denotes the results of our method with 25 uniformly sampled views.

Methods	Mean error (mm)
V2V (Moon, Chang, and Lee 2018)	15.57
AWR (Huang et al. 2020)	13.76
A2J (Xiong et al. 2019)	13.74
Rokid (Zhang et al. 2020)	13.66
Ours-1view	14.14
Ours-3views	13.24
Ours-9views	12.67
Ours-15views	**12.51**
Ours-25views	12.55

Table 2: Comparison with state-of-art methods on Hands19-Task1. We show the mean joint error on the test dataset split "Extrapolation", which aims to evaluating the model generalization performance and is the main evaluation metric on Hands19-Task1. "Ours-1view", "Ours-3views", "Ours-9views", "Ours-15views" and "Ours-25views" have the same meaning as shown in Table 1

Fig. 7 shows the qualitative comparison to other methods on a few testing examples of NYU dataset. Our method performs especially better on views with heavy occlusion and missing depth, e.g. the 3rd row. It is as expected since the re-render in a perpendicular virtual camera looking at the palm might be better to interpret the input depth.

We also compare our method with the state-of-the-art 3D hand pose estimation methods on Hands19-Task1. The details and results of state-of-the-art methods are cited from (Armagan et al. 2020). Rokid (Zhang et al. 2020) trains 2D CNN to regress joints with additional synthetic data. Compared to the used A2J backbone in our method, A2J (Xiong et al. 2019) reports results using higher resolution depth as input (384×384), deeper backbone (ResNet-152), and 10 backbone model ensemble. AWR (Huang et al. 2020) also provides results with model ensemble.

Table 2 shows the performance comparison. we can observe that our method with 3 selected views outperform other methods. Our method with 1 selected view performs slightly worse than AWR, A2J and Rokid, which can be due to using model ensemble in AWR and A2J and using additional synthetic training data in Rokid (Zhang et al. 2020).

More experimental results can be found in the supplementary document.
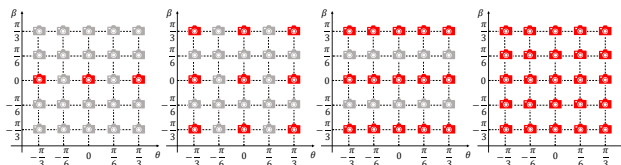
4.3 Ablation Study

Figure 8: Uniform sampling schemes. 25 virtual cameras are set uniformly. Red cameras are candidate virtual views. The figures from left to right show how 3, 9, 15, and 25 candidate virtual views are uniformly sampled.

Effect of View Number In order to investigate the effect of the number of virtual views with our method, we compare the hand pose performance on NYU dataset, ICVL dataset and Hands19-Task1 dataset by generating 3, 9, 15, 25 uniformly sampled virtual views ("UNIFOMR" column of Table 3). We also visualize the results of our method using different numbers of views on NYU dataset in Fig. 9 We observe that the hand pose estimation error decreases as the number of virtual views increases. Therefore, the virtual multi-view can boost the hand pose performance. The visualization results show that when using 25 views, the estimated wrist joints and joints in the missing depth value area are more accurate than using 3 views.

Effect of View Selection In this section, we investigate if the confidences are effective for view selection. We use our method to select 3, 9, 15 views from 25 candidate views and compare to uniform sampling as illustrated in Fig. 8. Though simple, the uniform sampling is actually a very strong strategy since it roughly guarantee at least 1 views close to some

Number of views	NYU			ICVL			Hands2019-Task1		
	UNIFORM	SELECT	LIGHT	UNIFORM	SELECT	LIGHT	UNIFORM	SELECT	LIGHT
1 view	7.93	7.23	7.34	5.56	5.18	5.16	14.39	14.03	14.14
3 views	7.14	6.73	6.82	5.27	4.85	4.86	13.67	13.07	13.24
9 views	6.77	6.43	6.53	4.96	4.77	4.77	12.81	12.60	12.67
15 views	6.55	6.38	6.41	4.85	4.76	4.76	12.61	12.51	12.51
25 views	6.40	-	-	4.79	-	-	12.55	-	-

Table 3: Comparison of mean joint error using uniform sampling and view selection on NYU, ICVL and Hands2019-Task1. "UNIFORM" denotes using uniformly sampled views. "SELECT" denotes selecting views from 25 uniformly sampled views with the "teacher" confidence network. "LIGHT" denotes selecting views from 25 uniformly sampled views with the "student" lightweight confidence network.
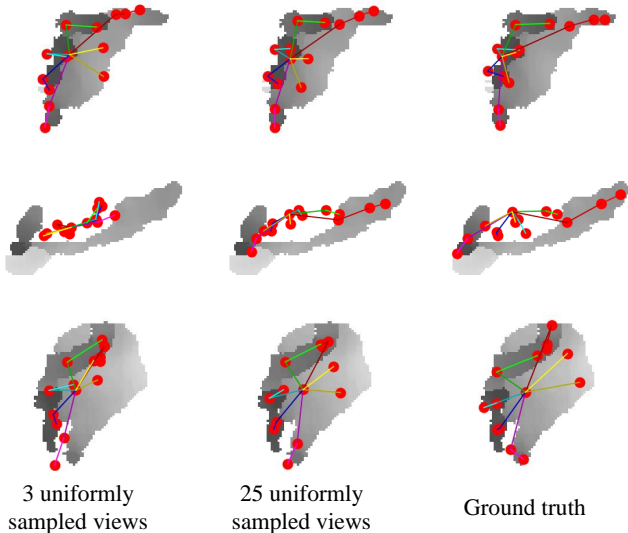

| 3 uniformly sampled views | 25 uniformly sampled views | Ground truth |

Figure 9: Comparison the visualization results of our method using different numbers of views on NYU dataset.

Number of views	UNIFORM	SELECT	LIGHT
1 view	61.57	27.92	47.43
3 views	56.71	28.19	46.42
9 views	41.56	27.61	39.48
15 views	36.50	27.95	34.57
25 views	26.71	-	-

Table 4: FPS comparison of uniform sampling and view selection on NYU dataset. "UNIFORM", "SELECT", "LIGHT" have the same meaning as shown in Table 3.

Component	NYU	ICVL	Hands2019-Task1
w confidence	6.40	4.79	12.55
w/o confidence	6.58	4.88	12.75

Table 5: Comparison of mean 3D joint error using different multi-view fusions on 25 uniformly sampled views.

good candidate. The results on NYU dataset, ICVL dataset and Hands19-Task1 dataset are shown in Table 3. For each experiment, we show the performance using teacher and student confidence network ("SELECT": teacher network, "LIGHT": student network). Our view selection consistently outperforms uniform selection in all the settings, which indicates that the predicted confidence is effective for view selection. Though using student network results in slightly worse performance than the teacher network, the overall computation cost is significantly reduced. As shown in Table 4, using student network almost doubles FPS. On ICVL dataset, the student network performs better than the teacher network, which may be due to low annotation quality and small scale of ICVL. More comparisons to random sampled views can be found in the supplementary document.

Comparison of Different Multi-View Fusion We now evaluate the confidence based fusion. In Table 5, we compare to direct average without the confidence on NYU dataset, ICVL dataset and Hands2019-Task1 dataset. The confidence based fusion achieves better performance on 3

datasets than direct average, which shows that the confidence is also beneficial to guide the fusion stage.

5. Conclusion

In this paper, we propose a deep learning network to learn 3D hand pose estimation from single depth, which renders the point cloud to virtual multi-view, and get more accuracy 3D hand pose by fusing the 3D pose of each view. Meanwhile, the confidence network we built uses multi-head attention to calculate the confidence of each virtual view. The confidence is used to improve the accuracy of 3D reconstruction during fusion and view selection. In order to obtain the confidence of each view efficiently, we obtain a lightweight confidence network using network distillation. Experiments on the main benchmark datasets demonstrate the effectiveness of our proposed method. Our method can also inspire several related researches such as scene parsing and reconstruction from depth, etc.

References

Armagan, A.; Garcia-Hernando, G.; Baek, S.; Hampali, S.; Rad, M.; Zhang, Z.; Xie, S.; Chen, M.; Zhang, B.; Xiong, F.; et al. 2020. Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction. In *Proceedings of European Conference on Computer Vision*, 85–101.

Deng, X.; Zhu, Y.; Zhang, Y.; Cui, Z.; Tan, P.; Qu, W.; Ma, C.; and Wang, H. 2020. Weakly supervised learning for single depth-based hand shape recovery. *IEEE Transactions on Image Processing*, 30: 532–545.

Erol, A.; Bebis, G.; Nicolescu, M.; Boyle, R. D.; and Twombly, X. 2007. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2): 52–73.

Fu, C.; Hu, Y.; Wu, X.; Wang, G.; Zhang, Q.; and He, R. 2021. High-fidelity face manipulation with extreme poses and expressions. *IEEE Transactions on Information Forensics and Security*, 16: 2218–2231.

Gärtner, E.; Pirinen, A.; and Sminchisescu, C. 2020. Deep Reinforcement Learning for Active Human Pose Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10835–10844.

Ge, L.; Cai, Y.; Weng, J.; and Yuan, J. 2018. Hand PointNet: 3D Hand Pose Estimation using Point Sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ge, L.; Liang, H.; Yuan, J.; and Thalmann, D. 2016. Robust 3D Hand Pose Estimation in Single Depth Images: from Single-View CNN to Multi-View CNNs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ge, L.; Ren, Z.; and Yuan, J. 2018. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision*, 475–491.

Huang, W.; Ren, P.; Wang, J.; Qi, Q.; and Sun, H. 2020. Awr: Adaptive weighting regression for 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11061–11068.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 28: 2017–2025.

Moon, G.; Chang, J. Y.; and Lee, K. M. 2018. 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/CVF Conference on Computer Vision and Pattern Recognition*, 5079–5088.

Pirinen, A.; Gärtner, E.; and Sminchisescu, C. 2019. Domes to Drones: Self-Supervised Active Triangulation for 3D Human Pose Reconstruction. *Advances in Neural Information Processing Systems*, 32: 3912–3915.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 652–660.

Rad, M.; Oberweger, M.; and Lepetit, V. 2018. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4663–4672.

Sun, X.; Wei, Y.; Liang, S.; Tang, X.; and Sun, J. 2015. Cascaded hand pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 824–832.

Tang, D.; Chang, H. J.; Tejani, A.; and Kim, T.-K. 2014. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3786–3793.

Tompson, J.; Stein, M.; Lecun, Y.; and Perlin, K. 2014. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5): 169.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wan, C.; Probst, T.; Van Gool, L.; and Yao, A. 2018. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5147–5156.

Xiong, F.; Zhang, B.; Xiao, Y.; Cao, Z.; Yu, T.; Zhou, J. T.; and Yuan, J. 2019. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 793–802.

Ye, Q.; Yuan, S.; and Kim, T.-K. 2016. Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. In *Proceedings of the European Conference on Computer Vision*, 346–361.

Zhang, Z.; Xie, S.; Chen, M.; and Zhu, H. 2020. HandAugment: A simple data augmentation method for depth-based 3D hand pose estimation. *arXiv preprint*.

Zhou, H.; Liu, J.; Liu, Z.; Liu, Y.; and Wang, X. 2020. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5911–5920.