Supplementary Material for "Point-Based Multi-View Stereo Network"

Rui Chen[1,3*] Songfang Han[2,3*] Jing Xu[1] Hao Su[3]
[1]Tsinghua University [2]The Hong Kong University of Science and Technology
[3]University of California, San Diego
chenr17@mails.tsinghua.edu.cn shanaf@connect.ust.hk
jingxu@tsinghua.edu.cn haosu@eng.ucsd.edu

1. Additional Ablation Study

1.1. Number of Point Hypotheses

In this section, we conduct an ablation study to verify the influence of the number of point hypotheses. In the main paper, we choose $m = 2$ for both the training and evaluation. We change to $m = 1$ and $m = 3$, and conduct the evaluation on the DTU evaluation set [1]. Table 1 shows the comparison result. Our proposed algorithm achieves best reconstruction quality in terms of completeness and overall quality when the number of point hypotheses is $m = 2$.

Point Hypotheses	Acc.(mm)	Comp.(mm)	Overall(mm)
1	**0.442**	0.515	0.479
2	0.448	**0.487**	**0.468**
3	0.468	0.499	0.484

Table 1: Ablation study of different number of point hypotheses m on the DTU evaluation set [1]. (The model is trained with $m = 2$.)

1.2. Number of Views

In this section, we study the influence of the number of input views N. Utilizing a variance-based cost metric, our Point-MVSNet can process an arbitrary number of input views. Although the model is trained using $N = 3$, we can evaluate the model using either $N = 2, 3, 5$ on the DTU evaluation set [1]. Table 2 demonstrates that the reconstruction quality improves with an increasing number of input views, which is consistent with common knowledge of MVS reconstruction.

Number of Views	Acc.(mm)	Comp. (mm)	Overall(mm)
2	0.462	0.604	0.533
3	0.448	0.507	0.478
5	**0.448**	**0.487**	**0.468**

Table 2: Ablation study on different number of input views N on the DTU evaluation set [1]. (The model is trained with $N = 3$)

2. Memory, runtime and overhead of *kNN*

Table 3 compares our memory usage and running speed against MVSNet. Our method is able to predict different resolutions of depth maps at different speed by changing the iterations. Naïve *kNN* of point cloud of N points can be memory-consuming with $O(N^2)$ complexity. However, we notice the *kNN* of a point tend to come from its nearby 2D pixels in the depth map. By leveraging this fact and taking the hypothetical points into consideration, we restrict the *kNN* search of each point from the whole point cloud to its $k \times k \times (2m + 1)$ neighborhood. Furthermore, we parallel the distance computation by using a fixed weight 3D kernel.

Iter.	Overall Err. (mm)	Resolution	GPU Mem. (MB)	Runtime (s)
0	0.726	160×120	7219	0.34
1	0.712	160×120	7221	0.61
2	0.468	320×240	7235	1.14
2†	0.474	320×240	7233	0.97
3	0.391	640×480	8731	3.35
MVSNet	0.551	288×216	10805	1.05

Table 3: Comparison of memory consumption and runtime. kNN is used for grouping, where all iterations adopt Euclidean distance, except for the iteration that is indicated by †, which uses pixel neighbor.

3. Post-processing

In this section, we describe the post-processing procedure in details. Similar to MVSNet [3], our post-processing is composed of three steps: photometric filtering, geometric consistency filtering and depth fusion.

For photometric filtering, we use predicted probability of the most likely depth layer as the confidence metric and filter out points whose confidence is below a threshold. The filtering threshold is set to 0.5 and 0.2 for coarse and our

[*] Equal contribution.

PointFlow stage, respectively. For geometric consistency, we calculate the discrepancy of predicted depths among multi-view predictions through reverse-projection. Points with discrepancy larger than 0.12mm are discarded. For depth fusion, we take average value of all reprojected depths of each point in visible views as the final depth prediction and produce the 3D point cloud.

4. Reconstruction Results

This section shows the reconstruction results of DTU dataset [1] and Tanks and Temples dataset [2] in Fig. 1 and Fig. 2 respectively. Point-MVSNet is able to reconstruct dense and accurate point clouds for all scenes.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, pages 1–16, 2016.

[2] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[3] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *European Conference on Computer Vision (ECCV)*, 2018.

Figure 1: Reconstruction results on the DTU evaluation set [1].

Figure 2: Reconstruction results on the intermediate set of Tanks and Temples [2].