Cost Volume Pyramid Based Depth Inference for Multi-View Stereo
—— Supplementary Material ——

Jiayu Yang[1], Wei Mao[1], Jose M. Alvarez[2], Miaomiao Liu[1,3]
[1]Australian National University, [2]NVIDIA, [3]Australian Centre for Robotic Vision
{jiayu.yang, wei.mao, miaomiao.liu}@anu.edu.au, josea@nvidia.com

In this supplementary material, we first introduce the multi-scale 3D convolutional network for cost volume regularisation in Section 1. In Section 2, we show additional qualitative reconstruction results by our method for different scans on the DTU dataset. In Section 3, we provide quantitative and qualitative evaluations on the reconstructed point cloud and depth map output of each level of the *Cost Volume Pyramid*. It shows that the reconstruction quality is improved on each iteration of depth residual refinement. In Section 4, we provide additional ablation studies analyzing the sensitivity to the feature dimension and the parameters of the fusion step on the final 3D reconstruction results.

1. 3D convolutional network architecture

The proposed CVP-MVSNet includes two convolutional neural networks modules: the *feature extraction network* and the multi-scale 3D convolutional network. The former is detailed in the submitted paper. In this section, we mainly provide more details regarding the multi-scale 3D convolutional network for cost volume regularisation.

Following the network structure in MVSNet [4], our cost volume regularisation network also has multiple scales and uses an encoder-decoder structure to maintain a large reception field while requiring less memory. However, different from MVSNet [4] which downsamples the input cost volume 3 times to form a 4-scale 3D CNN, we only use 2 scales in our architecture. In our case, the cost volume regularisation network is either used to estimate a coarsest depth map with small spatial resolution or search locally according to a upsampled depth map. In either case, our 2-scale 3D CNN provides sufficient neighbouring information for depth inference. More precisely, the 3D convolutional network, as shown in Fig. 1, consists of eight 3D convolutional layers to generate features of different scales and two deconvolutional layers to upsample those features. The architecture has skip connections from convolutional layers to deconvolutional layers, to form a 3-level UNet like structure [3].

The 3D CNN takes a regular 3D cost volume $\mathbf{C}^l \in \mathbb{R}^{W/2^l \times H/2^l \times M \times F}$ $l \in \{0, 1, \cdots, L\}$ as input and produces a probability volume $\mathbf{P}^l \in \mathbb{R}^{H/2^l \times W/2^l \times M}$ which indicates the probabilities of M depth (residual) hypothesis for each pixel at lth level.

iteration	depth map size	Acc.	Comp.	Overall	0.5mm *f-score*
4	1600×1152	**0.296**	**0.406**	**0.351**	**88.61**
3	800×576	0.340	0.418	0.379	86.82
2	400×288	0.632	0.561	0.596	64.34
1	200×144	1.02	0.910	0.966	30.17

Table 1: Quantitative results at different iterations. Additional refinement iterations leads to more accurate and complete point cloud. Qualitative results of these iterations are shown in Fig. 3

2. More results on DTU dataset

Fig. 2 shows additional results to those presented in the main paper. As shown, compared to the state of the art methods, our approach produces more accurate results. For instance, as shown in rows 1, 2, 4, 5 and 6, our method leads to more detailed geometrical structure. In addition, as shown in the third row, we obtain less noisy object surfaces.

3. Iterative refinement intermediate results

In Fig. 3, we show the point cloud fused with the depth map output for each level of the *Cost Volume Pyramid*. Moreover, in Fig. 4, we show intermediate depth map results. Table 1 provides a summary of quantitative results. As shown, the iterative depth map upsampling and depth residual refinement process improve the performance of our method both qualitatively and quantitatively.

4. Ablation study

4.1. Depth sampling for initial depth estimation

In the main paper, we have validated the influence of depth interval sampling on depth refinement while fixing the depth sampling for initial depth estimation. In this section, we show that it is also crucial to set a proper sampling interval for the coarsest depth map estimation. As shown in Table 2, using only 2-pixels leads to worse performance

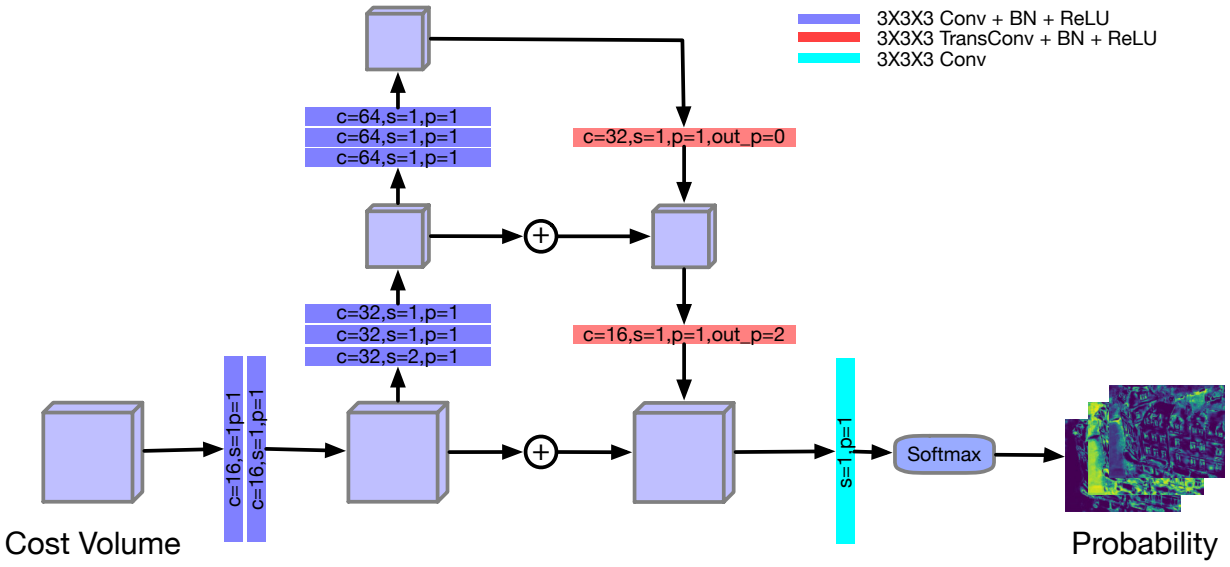


Figure 1: 3D ConvNet. Different from Point-MVSNet [1], we build a regular 3D cost volume on which a multi-scale 3D convolutional network is applied to estimate the probabilities of different depth (residual) hypotheses for each pixel.

depth interval	init. M	Acc.	Comp.	Overall	0.5mm *f-score*
2	12	0.421	0.941	0.681	67.66
1	24	0.316	0.406	0.361	88.07
0.5	48	0.308	**0.386**	**0.347**	**88.63**
0.25	96	**0.296**	0.406	0.351	88.61

Table 2: Results on DTU using different depth interval sampling for the coarsest depth estimation.

Feature CH	Acc.	Comp.	Overall	*f-score*	Mem.	Runtime
16	0.296	0.406	0.351	88.61	**8795**	**1.72**
32	**0.295**	**0.392**	**0.344**	**89.09**	14457	2.24

Table 3: Results comparison on DTU using image features of 16 and 32 channels.

as such a sparse sampling can not produce a good initial depth map for further refinement. On the other hand, using a dense sampling (e.g., 0.25 pixels) leads to a similar performance as using appropriate sampling interval (0.5-1 pixel). However, dense sampling consumes more memory since the cost volume becomes twice and four times larger compared to the memory consumption of using 0.5 pixel and 1 pixel, respectively.

4.2. Feature channel size

Similar to state-of-the-art methods [1, 4, 5], we first extract image features using a 2D convolutional network (*feature extraction network*). However, different from related approaches that require more than 32 channels to extract features [1, 4, 5], we empirically observe that our method is stable with respect to the number of channels. In the main paper, we showed that a 16-channel image feature is sufficient to produce better point cloud fusion than state-of-the-art methods [1, 4, 5] on DTU dataset. We further provide the performance of using image feature with 32 channels. As shown in Table 4, double the channel size only improves the *f-score* for 0.5%, but with 64% and 30% increase in memory usage and runtime accordingly.

input size	Acc.	Comp.	Overall	*f-score*	Mem.	Runtime
400×288	0.632	0.561	0.596	64.34	**537**	**0.155**
800×576	0.629	0.548	0.588	65.94	674	0.210
1600×1152	**0.626**	**0.537**	**0.581**	**66.64**	1071	0.436

Table 4: Results on DTU using input images of different size to generate depth map of the same resolution (400×288). Using a larger input image provides slightly better performance but consumes much more memory and is significantly slower.

4.3. Depth map size VS. image size

A high advantage of our method is providing depth map of the same resolution as the input image. It differs from previous methods [1, 4, 5] which usually require input image of high resolution in order to generate depth maps that are 2 times or 4 times as small as the input image resolution.

To verify that image provides enough information to estimating depth map of the same size reliably, we train our model to generate depth map of the same resolution (400×288) with input images of different resolution. Note that since the final output depth map size is the same, the number of iteration for refinement is kept the same (3 times) as experiments with input image size of different resolution. The results are shown in Table 4. Using larger input image consumes much more memory, is slower but only has slight improvements on the final results.

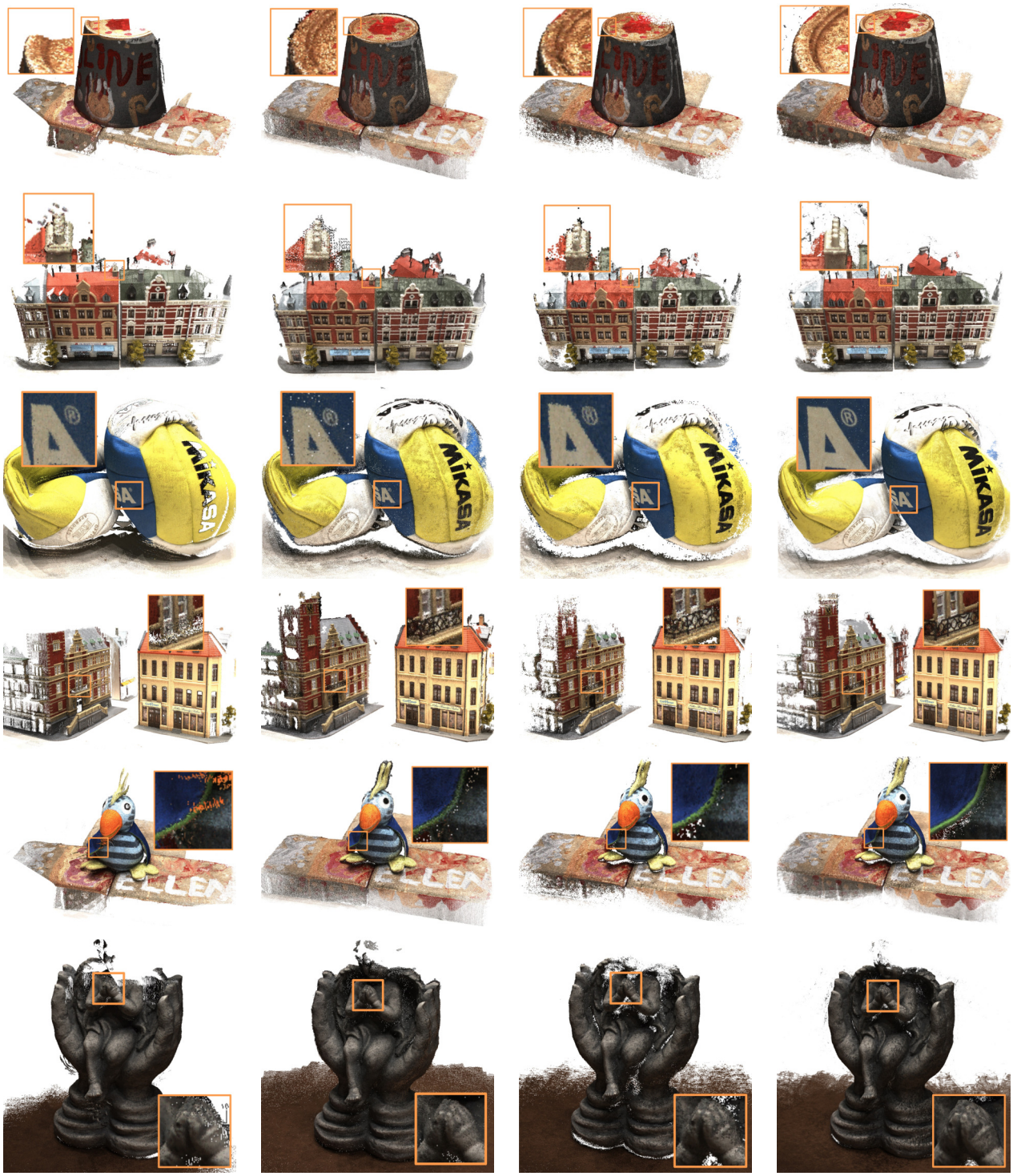
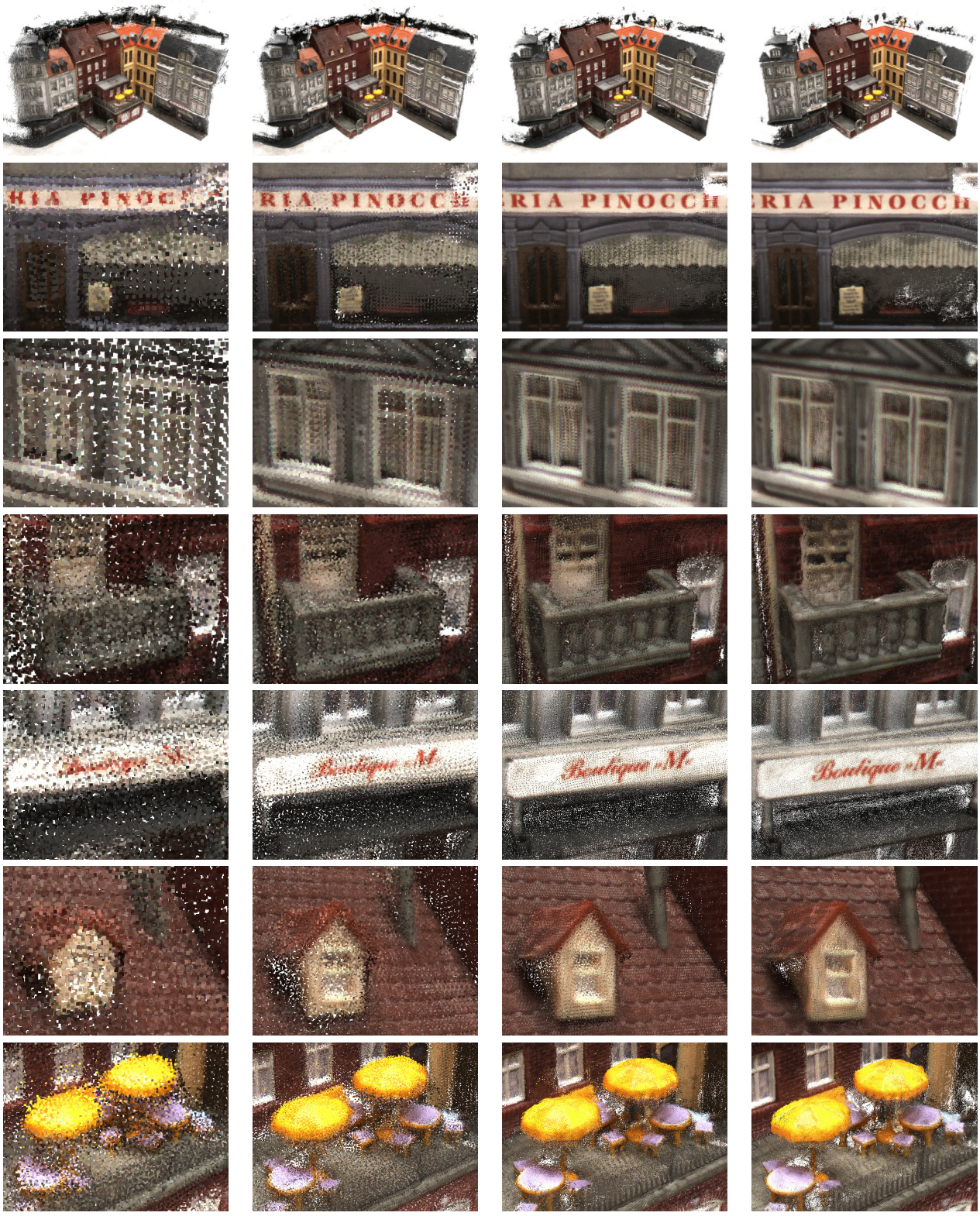


Figure 2: More qualitative results on DTU dataset. Best viewed on screen. Our method produces more accurate and detailed geometrical structure (row 1,2,4,5,6) and less noisy object surface (row 3)



Iteration 1 Iteration 2 Iteration 3 Iteration 4

Figure 3: Intermediate point cloud results. Our method provide improve details for every iteration of depth residual refinement. Best viewed on screen.

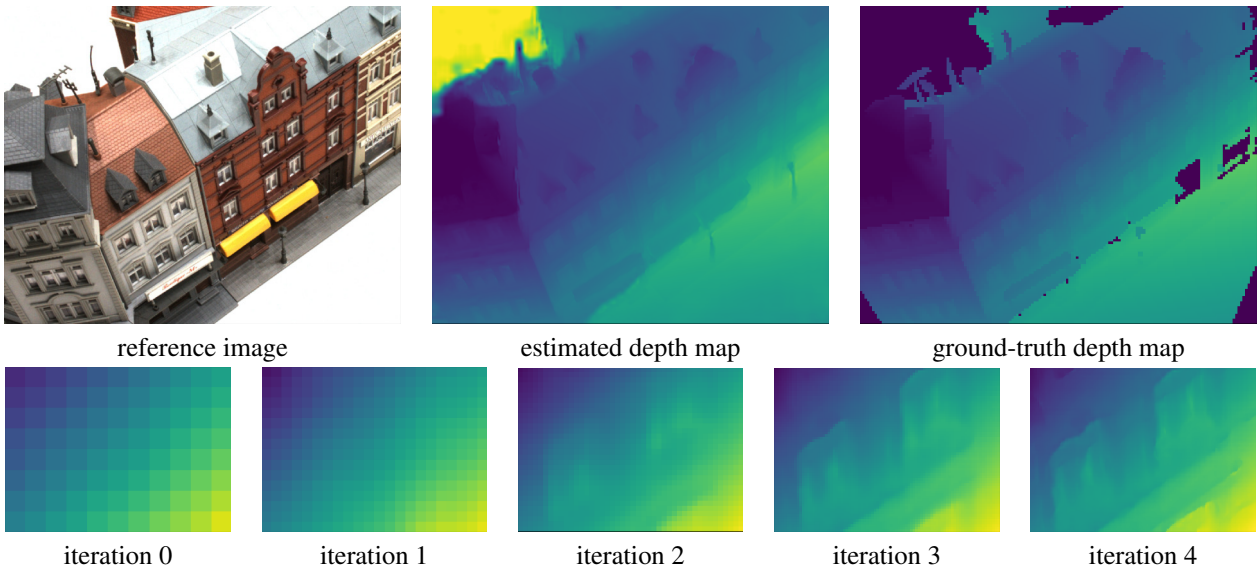


Figure 4: Depth map results on DTU dataset. Best viewed on screen. Bottom row shows an area of the intermediate depth map produced by each iteration of depth residual refinement. Details are improved on every iteration of depth residual refinement.

threshold	Acc.	Comp.	Overall	0.5mm *f-score*
0.20	0.331	**0.357**	**0.344**	88.55
0.15	0.307	0.388	0.347	**88.68**
0.13	0.296	0.406	0.351	88.61
0.10	0.279	0.443	0.361	88.21
0.05	**0.245**	0.590	0.418	85.14

Table 5: Results on DTU for different geometric consistency thresholds. The trade-off between accuracy and completeness is generally ruled by the filtering threshold.

4.4. Fusion parameters and trade-off

We study the influence of parameters used in the point fusion method, namely Fusibile [2], on the depth map results. In particular, we validate the geometry consistency threshold setting with $\{0.2, 0.15, 0.1, 0.05\}$. Results are shown in Table 5. As shown, a lower threshold leads to better accuracy at the expense of completeness. In our paper, results are obtained setting the threshold to 0.13.

References

[1] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1538–1547, 2019. 2, 3

[2] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V*, 25:361–369, 2016. 5

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[4] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 1, 2

[5] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019. 2, 3