MVPSNet: Fast Generalizable Multi-view Photometric Stereo: Supplementary Material

1. Overview

In this supplementary material, we will include the following contents:

- We describe more details about our **sMVPS dataset** in **Section 2** and show additional example images in Figure 1 and Figure 2.
- We provide additional **experiment details**, including notations we use for network architecture and implementation details in **Section 3**.
- We explain our **mesh extraction pipeline** in detail in **Section 4** together with the parameters we use.
- We provide the equations of the **evaluation metrics** we use in **Section 5**.
- In the main paper, we provide L1 Chamfer distance and F-score with L2 distance after ICP [3, 4, 5, 25]. Here in **Section 6**, we also provide **results of L1 Chamfer distance and F-score with L2 distance before ICP** in Table 1 and 2.
- Comparison between pretrained MVS models (CasMVSNet [8] and TransMVSNet [6]) and the models retrained on our sMVPS dataset in Table 3.
- We include additional qualitative results. We show the global shape of reconstructed mesh from each method under three different views in Figure 3 - 7. We also show additional zoomed areas for visual comparison between meshes in Figure 8.

2. sMVPS datasets

Object and Camera Positioning For both sMVPS-sculpture and sMVPS-random datasets objects are placed at the center of the world coordinate system with the object's *up* direction along the z-axis. Objects are scaled to be inside a sphere of radius one. We use a pinhole camera for rendering with an FOV of 9.3°, which is similar to the FOV used to capture the DiLiGenT-MV dataset [18]. Camera positions are most easily described in spherical coordinates, i.e. an azimuth angle, a polar angle, and a radial distance. The azimuth angle for the ith camera is $(18 + X_i)^\circ$ where X_i is a uniform random number between -3 and 3, and i runs from 0 to 19. The polar angle for each camera is sampled uniformly from 62° - 64°. The radial distance is sampled uniformly between 14 and 16.5. This distance is chosen so the object occupies the majority of the image.

Light Positioning Each view is rendered under 10 directional lights. The first light is always co-directional with the camera while the other 9 are randomly sampled from the spherical cap centered on the camera's optical axis with an angle of 45°.

BRDF To generate BRDFs we follow [19]. Namely we use the Cook-Torrance BRDF model with spatially-varying albedo drawn from 415 free textures from [1], and randomly generate roughness as described in [19]. Roughness is constant in the case of sMVPS-sculpture and constant for each primitive in the case of sMVPS-random.

Object Meshes For the sMVPS-random dataset objects are drawn from the collections of random primitives generated by [23] using a 90-10 train/test split. For the sMVPS-sculpture dataset we use the following meshes from [22] to render the training set: nymphe-seated, standing-isis-priest, the-slave-girl, thor, three-danish-polar-explorers, tiger-devouring-a-gavial, two-wrestlers-in-combat,ugolino-and-his-sons, virgin-and-child, woman-associated-with-the-cult-of-isis, wounded-amazon, wounded-cupid, wrestling-decimated-cleaned and the mesh virgin-mary-with-her-dead-son for the test set.

Rendering Images are rendered with Mitsuba 2 using the path-tracer integration method. We render at a resolution of 612x512 with 128 samples-per-pixel.

More Examples To further show the diversity on surface shapes, textures and materials of our sMVPS dataset, we provide additional example images of sMVPS-sculpture in Figure 1 and sMVPS-random in Figure 2.

3. Additional experiment details

3.1. Notation in Figure 1 of main paper

The architecture of our network is illustrated in Figure 1 in the paper and we describe a few details and the notations we use:

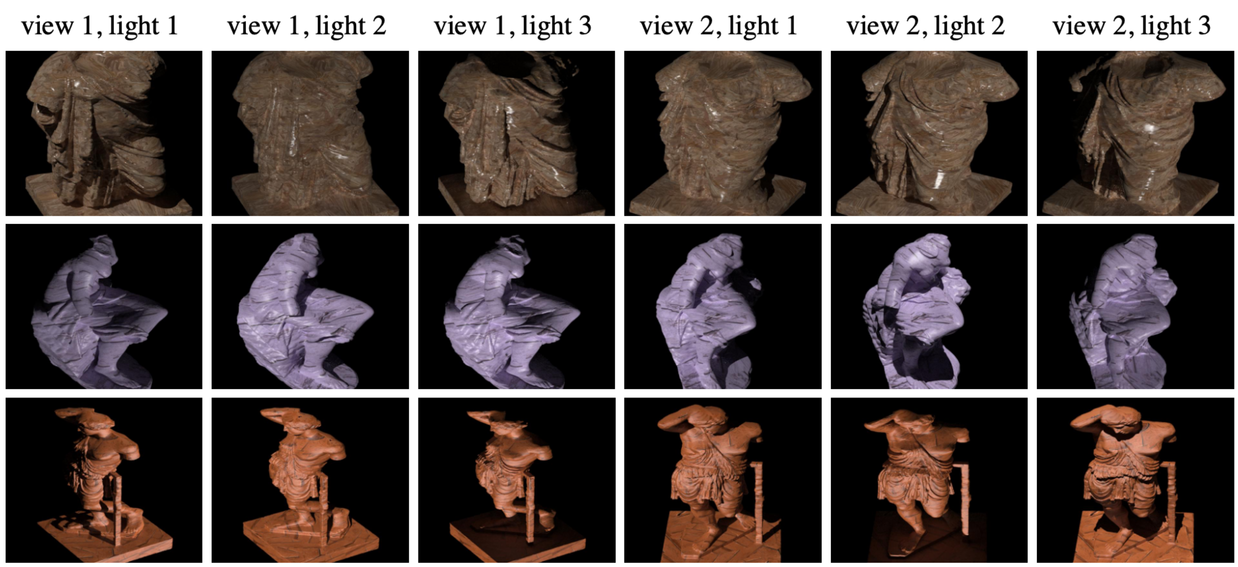

Figure 1. Additional example images of sMVPS-sculpture.

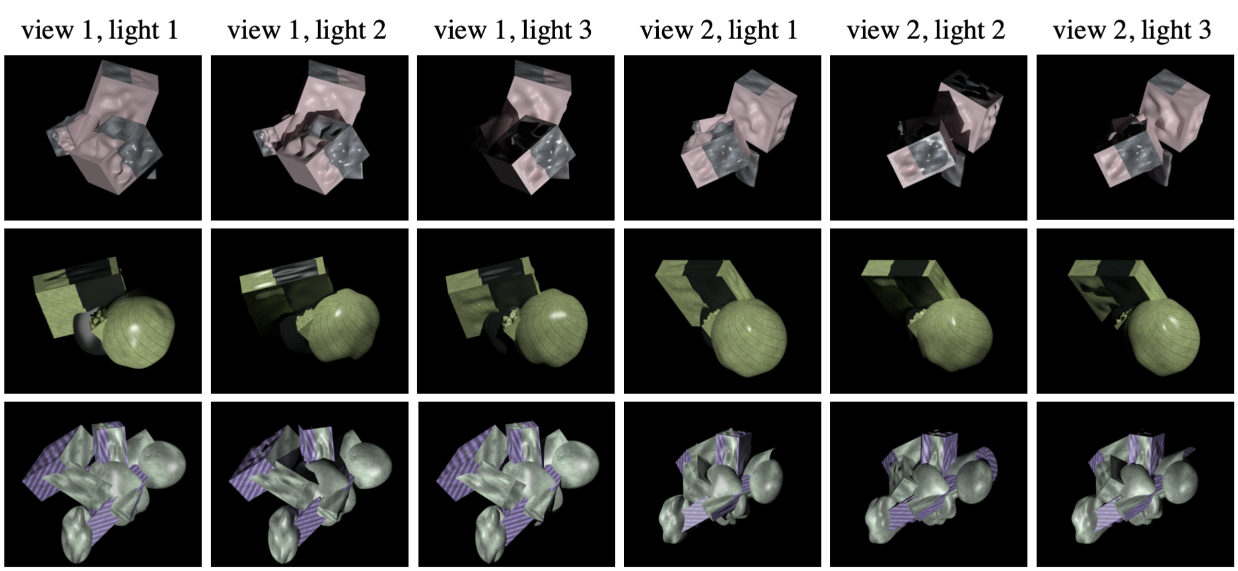

Figure 2. Additional example images of sMVPS-random.

ResBlk: Resnet block. It consists of *conv2d(kernel=3)* $\rightarrow$ *BatchNorm* $\rightarrow$ *ReLu* $\rightarrow$ *conv2d(kernel=3)* $\rightarrow$ *BatchNorm*. And the input of this block is added to the output of this block as a residual connection [10].

Tconv: ConvTranspose2d layer in Pytorch with kernel=3.

3.2. Implementation details

Our model is implemented in Pytorch [21] and we use a NVIDIA RTX A6000 GPU to train it. For input images, we crop them to 512×512 and rescale the pixel values to (0, 1). For each training sample, we use 3 views and 3 lightings. It is challenging to find correspondences for view selection in textureless regions, so we simply take the two adjacent views of a reference view as source views. To make our model more robust to different lighting configurations, we randomly sample 3 lightings and use the same lightings for all views, resulting in $3 \times 3 = 9$ images for each training sample. We use Adam [16] optimizer and set betas as $(0.9, 0.999)$. We trained 50 epochs in total. The initial learning rate is 0.001 and it decays to half at steps [8, 12, 30, 40]. To get ground truth depth map of DiLiGenT-MV [18], we render depth map from ground truth mesh and camera parameters.

4. Mesh extraction pipeline

We use the same mesh extraction pipeline to recover 3D mesh from predicted depth maps for all methods for a fair comparison.

4.1. Depth filtering

We use two kinds of masks to filter predicted depth maps. First, we employ 2D object masks to rule out background. This is because our model is only trained on pixels within an object. Second, we apply geometric filtering to only keep depth predictions that are consistent across adjacent views. For each object pixel in the reference view, p_r, we have a predicted depth aligned with this view d_r. We lift p_r to a 3D point P_r and project P_r to a source view pixel p_s. Assume the predicted depth of source view at p_s is d_s. By lifting p_s using d_s, we get a 3D point P_s. Projecting P_s back to the reference view gives us a reprojected pixel p_r' and a depth d_r'. We set thresholds for the distance between the original pixel p_r and the reprojected pixel p_r' as well as relative difference between d_r and d_r' as follows:

$$dist(p_r, p_r') < 1, \tag{1}$$

$$abs(d_r - d_r')/d_r < 0.01 \tag{2}$$

For each pixel p_r and its corresponding depth estimation d_r, we check this geometric consistency with each source view and keep them only if the consistency holds for at least one source view.

4.2. Depth fusion

After the depth filtering step, we combine each depth map in a fusion step. For an object pixel p_r, we simply average over d_r and all the estimations from source views that are consistent with it, d_{s_i} for $i = 1, ..., i_N$, where i_N is the total number of geometric consistent neighboring views, and use this average as depth at p_r. We then lift p_r to a vertex in point cloud and attach the predicted normal, n_r, to it. This way, we get point cloud utilizing information from all views.

Note there are other possible depth fusion methods, *e.g.* GIPUMA [7], some of which may achieve better fusion performance for certain datasets. But there is no method that is better for all datasets, so we leave exploration in this direction as a future work.

4.3. Surface reconstruction

We apply Screend Poisson Surface Reconstruction (SPSR) [15] to recover mesh from point cloud. We use same set of parameters for all methods and all objects. Specifically, we set $reconstruction\_depth = 8$, $minimum\_number\_of\_samples = 1.5$ and $interpolation\_weight = 4$. Note that before recovering surfaces, an extra step of computing normal based on the point cloud is needed for methods without normal prediction, i.e., CasMVSNet [8] and TransMVSNet [6].

5. Evaluation metrics details

We use L1 Chamfer distance (mm) and F-socre with L2 distance (threshold at 1mm) to evaluate the quality of the reconstructed mesh. Both metrics are applied to two sets of 3D points, which are vertices of the reconstructed mesh and the ground truth mesh.

Give two point sets, $\mathcal{R}$ and $\mathcal{G}$, L1 Chamfer distance is defined as follows:

$$CD(\mathcal{R}, \mathcal{G}) = \frac{1}{|\mathcal{R}|}\sum_{x\in\mathcal{R}}\min_{y\in\mathcal{G}}\|x-y\| + \frac{1}{|\mathcal{G}|}\sum_{y\in\mathcal{G}}\min_{x\in\mathcal{R}}\|x-y\|. \tag{3}$$

We use the F-score similarly defined as [17]. For a reconstructed point $r \in \mathcal{R}$, its L2 distance to the ground truth mesh $\mathcal{G}$ is

$$e_{r\to\mathcal{G}} = \min_{g\in\mathcal{G}}\|r-g\|_2, \tag{4}$$

and for a ground truth point $g \in \mathcal{G}$, its distance to the reconstructed mesh is defined as:

$$e_{g\to\mathcal{R}} = \min_{r\in\mathcal{R}}\|r-g\|_2, \tag{5}$$

The precision and recall for a threshold d are:

$$P(d) = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}[e_{r\to\mathcal{G}} < d] \tag{6}$$

$$R(d) = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}[e_{g\to\mathcal{R}} < d] \tag{7}$$

F-score is the harmonic mean of precision and recall as a summary measure:

$$F(d) = \frac{2P(d)R(d)}{P(d)+R(d)} \tag{8}$$

6. Additional results without ICPs

In the paper, we report results after ICP [3,4,5,25], which is an extra registration step we applied to all meshes after being reconstructed. It is initially aimed to fairly compare our mesh with others as several methods indicate that they did registration after extracting meshes [14, 18]. We find it helpful to improve accuracy of several methods, even for some of those that already have registration applied. Since there is no standard way to do registration among existing methods, we applied ICP to meshes from all methods, regardless of whether they have done registration or not.

For a complete comparison, we also provide the quantitative results of L1 Chamfer distance and F-score with L2 distance (threshold at 1mm) without ICP [3, 4, 5, 25] in Table 1 and Table 2, respectively. They show that even without registration, our method can still perform comparably with state-of-the-art methods with registration.

7. Effectiveness of sMVPS dataset

We provide L1 Chamfer distance in mm and F-score with L2 distance (threshold at 1mm) of pretrained CasMVSNet [8] and TransMVSNet [6] together with the models trained using our sMVPS dataset in Table 3, which further demonstrates the effectiveness of the proposed sMVPS dataset.

References

[1] 3d textures. https://3dtextures.me/. Accessed: 2020. 1

[2] 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*, 120:153–168, 2016. 6

[3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987. 1, 3, 4

[4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. 1, 3, 4

[5] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 1, 3, 4

[6] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8585–8594, 2022. 1, 3, 4, 6

[7] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015. 3

[8] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. 2019. 1, 3, 4, 6

[9] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020. 5

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[11] Satoshi Ikehata. Ps-transformer: Learning sparse photometric stereo network using self-attention mechanism. *arXiv preprint arXiv:2211.11386*, 2022. 5

[12] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncertainty-aware deep multi-view photometric stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12601–12611, 2022. 5

[13] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Multi-view photometric stereo revisited. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3126–3135, 2023. 5

[14] Berk Kaya, Suryansh Kumar, Francesco Sarno, Vittorio Ferrari, and Luc Van Gool. Neural radiance fields approach to deep multi-view photometric stereo. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1965–1977, 2022. 3, 5

[15] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 3

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[17] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 3

[18] Min Li, Zhenglong Zhou, Zhe Wu, Boxin Shi, Changyu Diao, and Ping Tan. Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE Transactions on Image Processing*, 29:4159–4173, 2020. 1, 3, 5, 6, 7, 8, 9

[19] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. Shape and material capture at home. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6123–6133, 2021. 1

[20] Jaesik Park, Sudipta N Sinha, Yasuyuki Matsushita, Yu-Wing Tai, and In So Kweon. Robust multiview photometric stereo using planar mesh parameterization. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1591–1604, 2016. 5

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 2

[22] Olivia Wiles and Andrew Zisserman. Silnet : Single- and multi-view reconstruction by learning from silhouettes. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017. 1

[23] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4):126, 2018. 1

	Per-scene optimization					Generalizable		
Category	Manual Effort		Standalone			Single-view PS	MVS	MVPS
Method	PJ16 [20]	LZ20 [18]	BKW22 [14]	BKC22 [12]	PS-NeRF [24]	PS-Transformer [11]	CasMVSNet [9]- RT	Ours
BEAR	2.63	0.74	1.03	1.09	**0.81**	3.25	1.38	<u>0.91</u>
BUDDHA	1.18	0.99	2.44	1.19	**0.98**	4.44	1.30	<u>1.12</u>
COW	1.16	0.39	1.08	0.86	**0.78**	2.67	1.26	<u>0.80</u>
POT2	3.27	0.69	1.32	1.32	**0.81**	2.92	1.43	<u>0.94</u>
READING	1.49	0.74	1.94	0.93	0.98	3.69	<u>0.83</u>	**0.76**
AVERAGE	1.95	0.71	1.56	1.08	**0.87**	3.39	1.24	<u>0.91</u>

Table 1. L1 Chamfer Distance in mm (lower is better) between reconstructed mesh and GT without ICP. '-RT' denotes trained on our synthetic MVPS dataset. For non-manual methods, the best result is shown in bold, 2nd best as underline. LZ20 & PJ16 involve carefully crafted steps, manual efforts in finding correspondence, and an initial mesh or point cloud.

	Per-scene optimization						Generalizable		
Category	Manual Effort		Standalone				Single-view PS	MVS	MVPS
Method	PJ16 [20]	LZ20 [18]	BKW22 [14]	BKC22 [12]	BKW23* [13]	PS-NeRF [24]	PS-Transformer [11]	CasMVSNet [9]-RT	Ours
BEAR	0.504	0.987	0.926	0.895	0.965	**0.994**	0.496	0.902	<u>0.990</u>
BUDDHA	0.935	0.935	0.745	0.922	**0.993**	<u>0.970</u>	0.387	0.913	0.953
COW	0.917	0.990	0.943	0.981	<u>0.987</u>	0.984	0.617	0.896	**0.993**
POT2	0.459	0.985	0.929	0.909	<u>0.991</u>	0.990	0.609	0.891	**0.992**
READING	0.868	0.975	0.807	0.970	0.975	0.946	0.501	<u>0.981</u>	**0.989**
AVERAGE	0.737	0.974	0.870	0.935	<u>0.982</u>	0.977	0.522	0.917	**0.983**

Table 2. F-score on L2 distance (higher is better) between reconstructed mesh and GT without ICP. '-RT' denotes trained on our synthetic MVPS dataset. For non-manual methods, the best result is shown in bold, 2nd best as underline. LZ20 & PJ16 involve carefully crafted steps, manual efforts in finding correspondence, and an initial mesh or point cloud. BKW23* code not available, result from the paper.

[24] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K Wong. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. *arXiv preprint arXiv:2207.11406*, 2022. 5, 10

[25] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994. 1, 3, 4

Metrics	L1 Chamfer distance					F-score (1mm)				
Method	CasMVSNet [8]	CasMVSNet-RT	TransMVSNet [6]	TransMVSNet-RT	Ours	CasMVSNet [8]	CasMVSNet-RT	TransMVSNet [6]	TransMVSNet-RT	Ours
BEAR	2.00	1.47	1.02	1.48	**0.80**	0.789	0.911	0.962	0.882	**0.991**
BUDDHA	1.44	1.26	1.09	1.10	**1.07**	0.878	0.919	0.961	0.963	**0.958**
COW	2.73	1.27	1.15	1.05	**0.77**	0.658	0.914	0.927	0.941	**0.993**
POT2	1.89	1.46	1.10	1.05	**0.82**	0.799	0.901	0.956	0.964	**0.994**
READING	1.07	0.75	0.87	0.76	**0.66**	0.941	0.980	0.971	0.978	**0.988**
AVERAGE	1.83	1.24	1.05	1.09	**0.82**	0.813	0.925	0.955	0.946	**0.985**
Recon. Time/object	22s	22s	52s	52s	105s	22s	22s	52s	52s	105s

Table 3. Results of CasMVSNet and TransMVSNet on L1 Chamfer distance in mm and F-score with L2 distance (threshold at 1mm) after ICP. CasMVSNet [8] and TransMVSNet [6] denote the pretrained models on DTU dataset [2]. 'RT' denotes trained on our synthetic MVPS dataset.

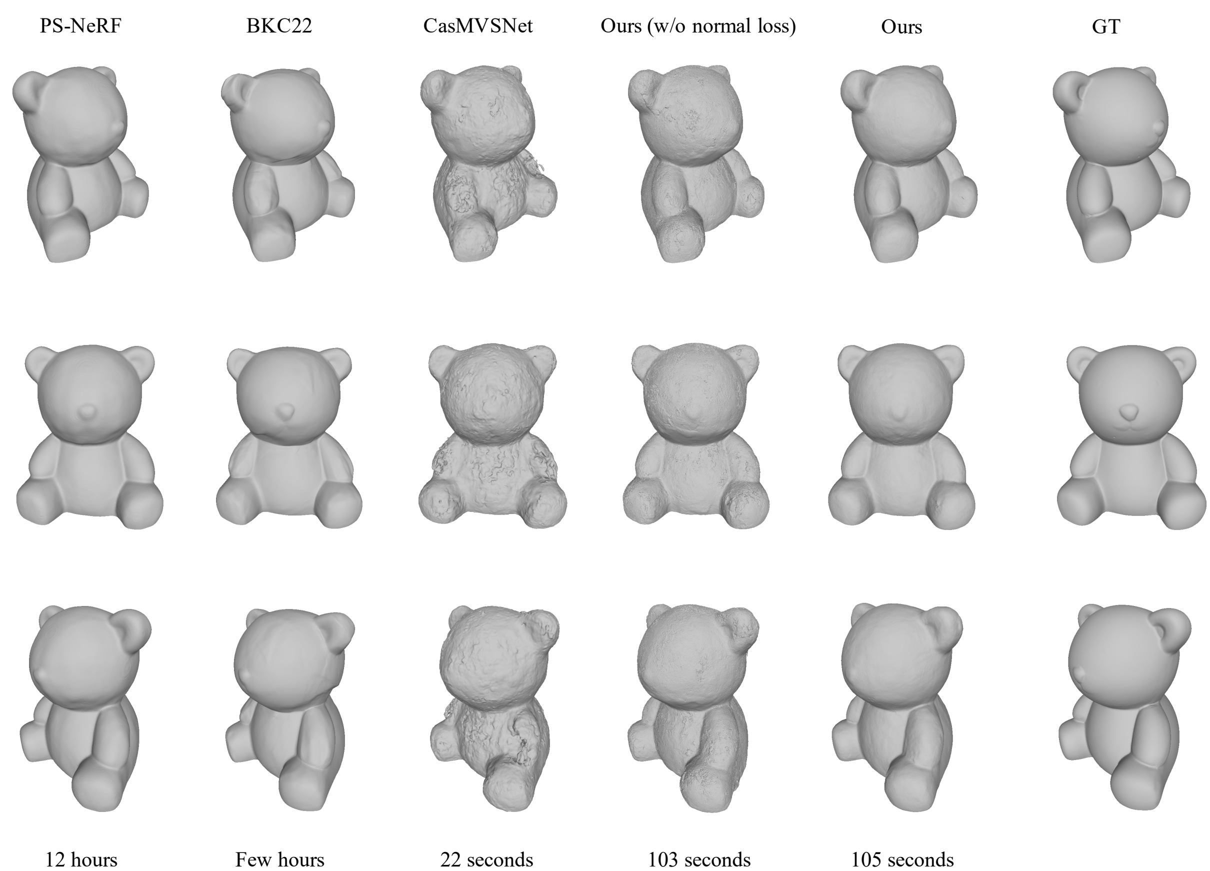

Figure 3. Reconstruction of BEAR under three different views (left-side, front, right-side) in DiLiGenT-MV [18]. Last row is reconstruction time.

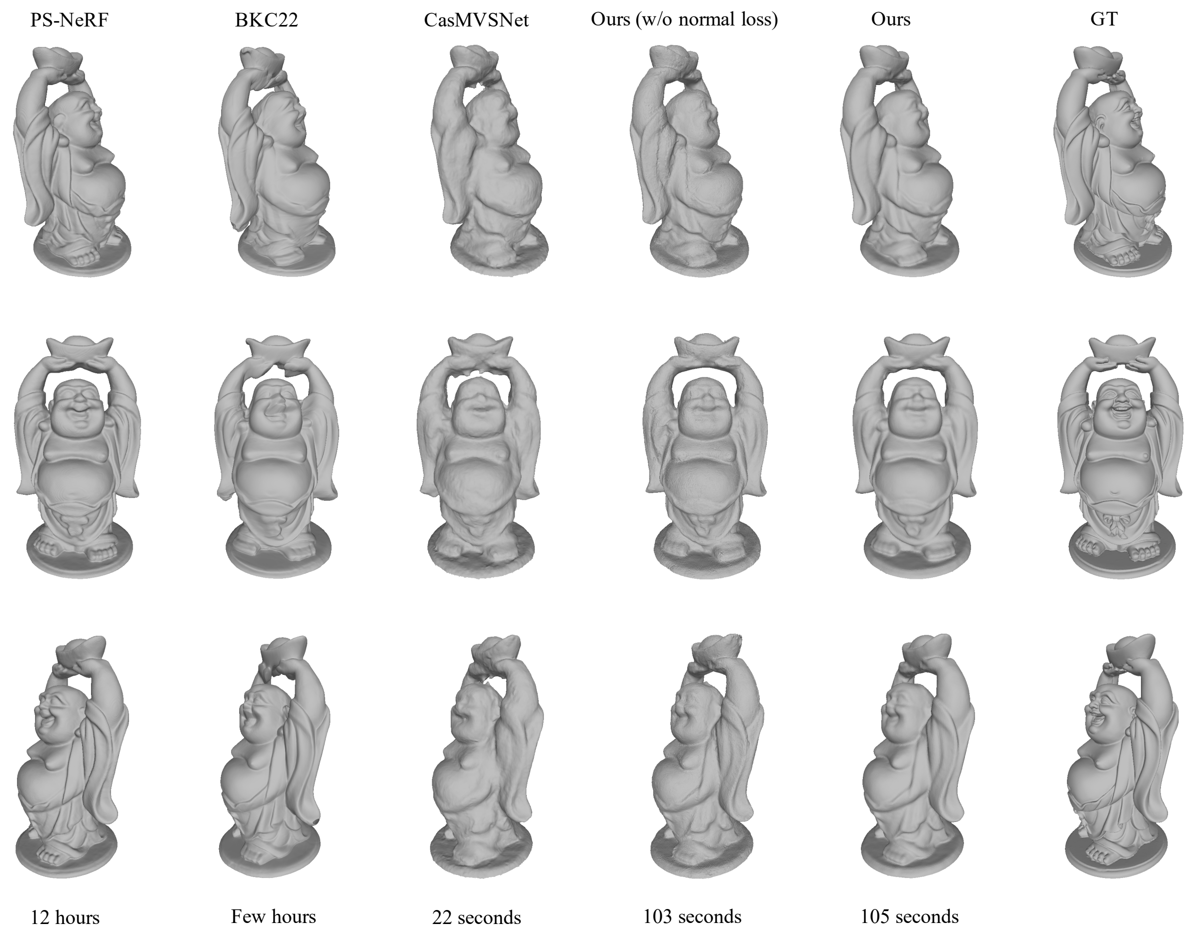

Figure 4. Reconstruction of BUDDHA under three different views (left-side, front, right-side) in DiLiGenT-MV [18]. Last row is reconstruction time.

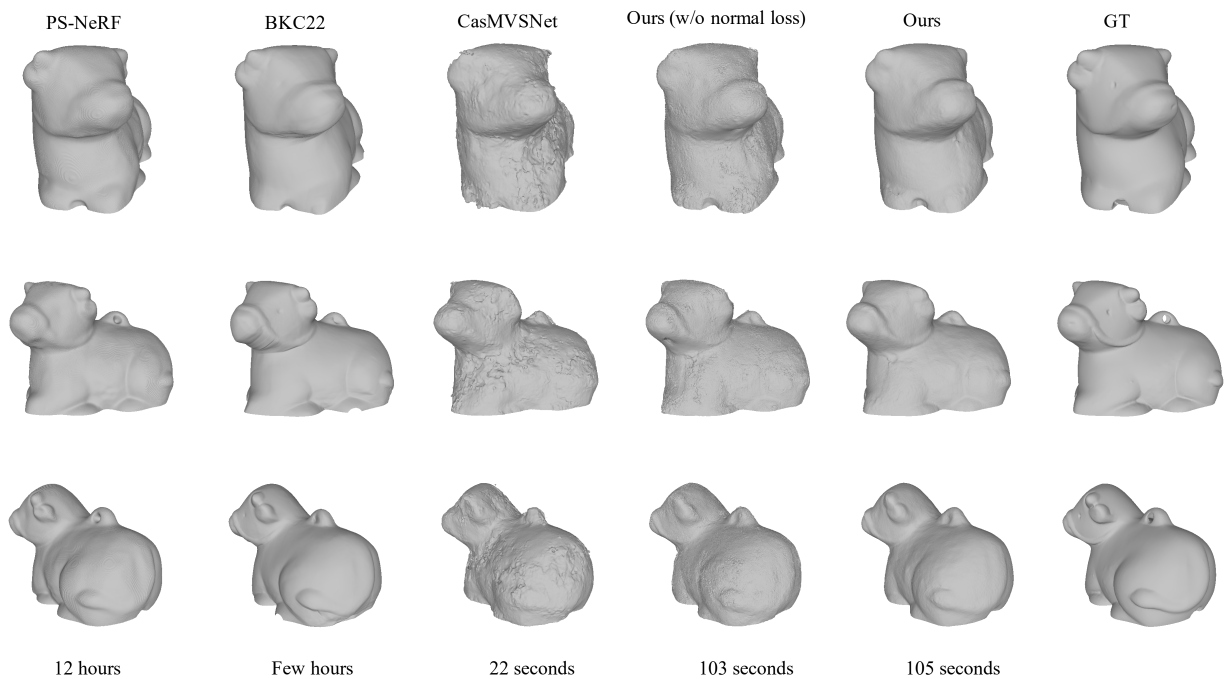

Figure 5. Reconstruction of COW under three different views (front, right-side, back) in DiLiGenT-MV [18]. Last row is reconstruction time.

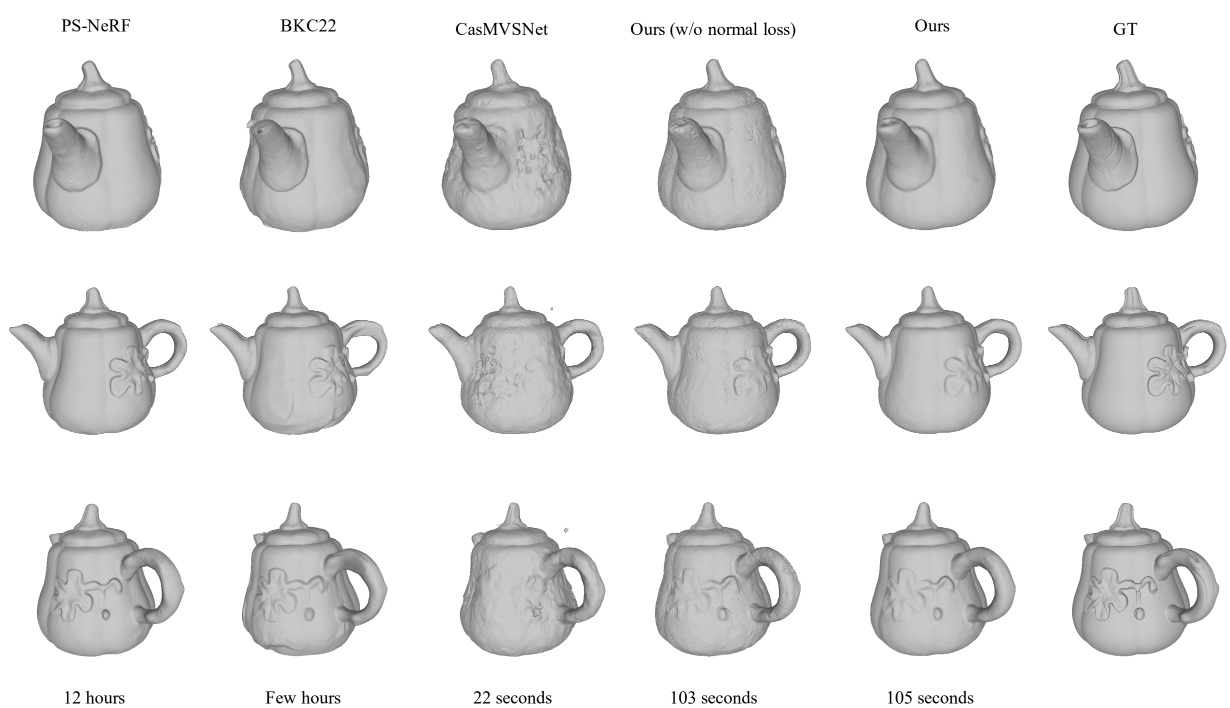

Figure 6. Reconstruction of POT2 under three different views (left-side, front, right-side) in DiLiGenT-MV [18]. Last row is reconstruction time.

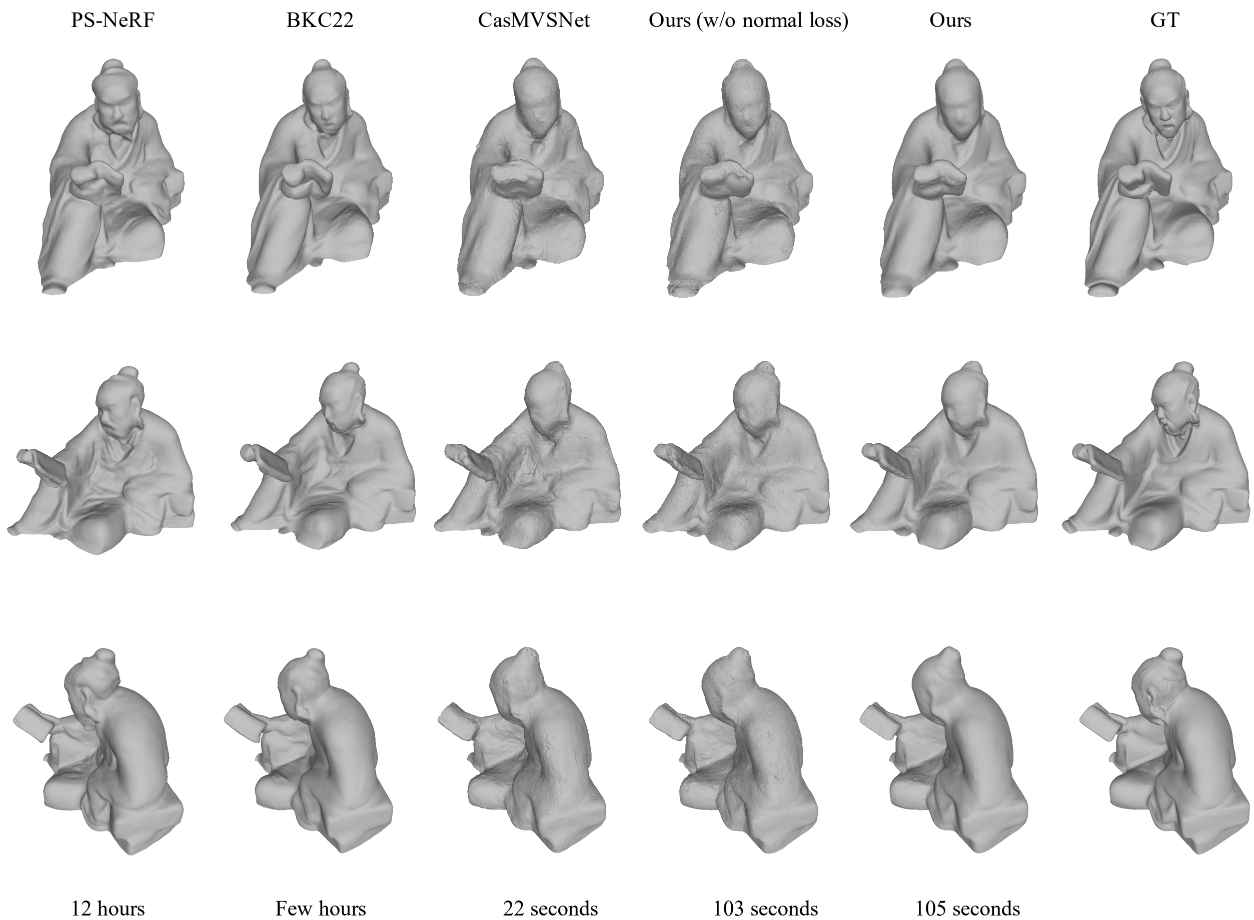

Figure 7. Reconstruction of READING under three different views (front, right-side, right) in DiLiGenT-MV [18]. Last row is reconstruction time.

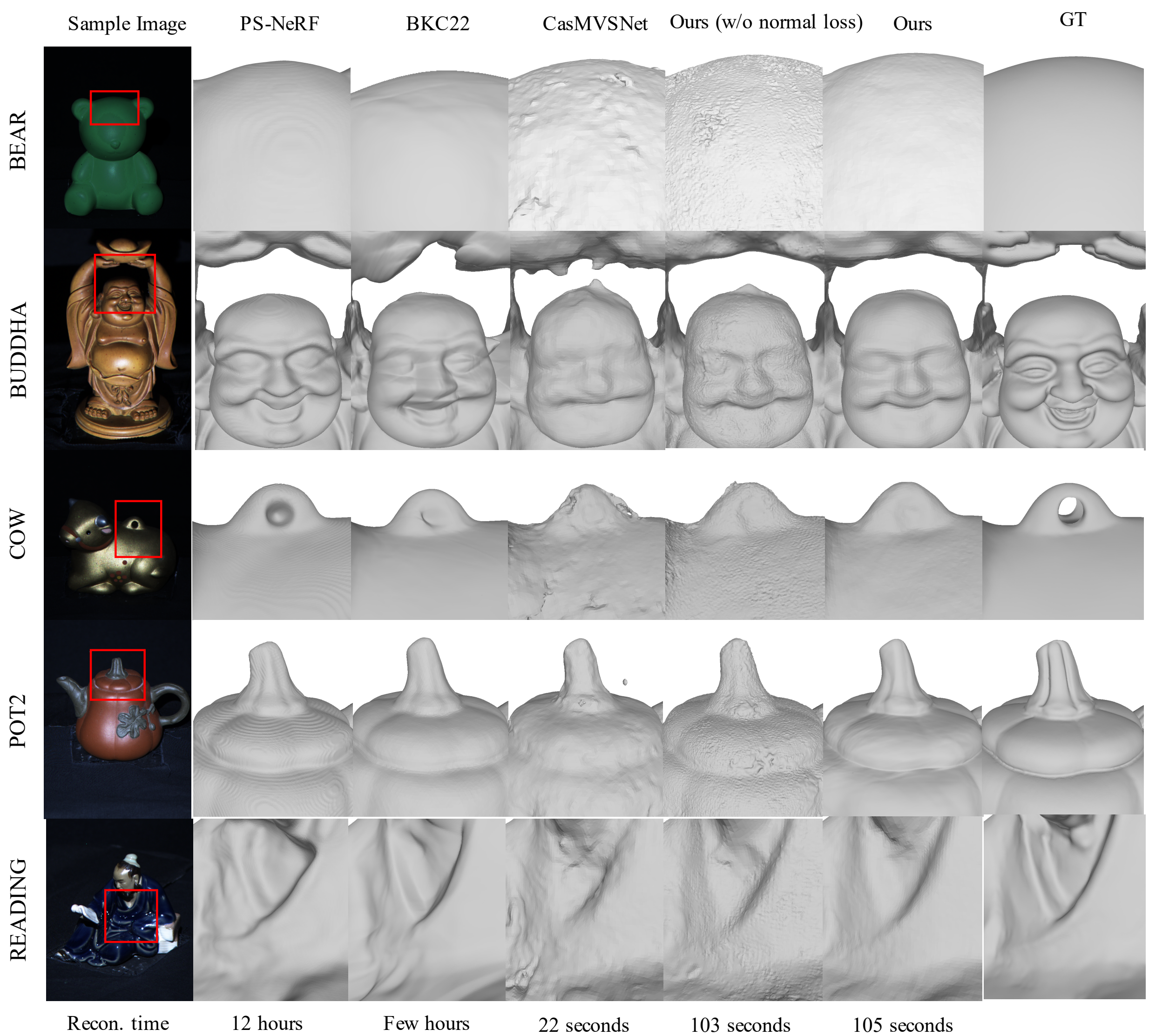

Figure 8. Zoomed-in areas on meshes from all methods. We observe that in general PS-NeRF [24] provides meshes with find details while it often contains iso-contour pattern artifacts. Our method can provide smooth meshes with correct global shapes even though it takes very short time compared with other methods.