PS-NeRF: Neural Inverse Rendering for Multi-view Photometric Stereo (Supplementary Material)

Wenqi Yang[1] Guanying Chen[2⋆] Chaofeng Chen[3]
Zhenfang Chen[4] Kwan-Yee K. Wong[1]

[1]The University of Hong Kong [2]SSE and FNii, CUHK-Shenzhen
[3]Nanyang Technological University [4]MIT-IBM Watson AI Lab

1 More Details for the Proposed Method

1.1 More Details for Uncalibrated Photometric Stereo

We adopted a recent uncalibrated photometric stereo (UPS) method, called SDPS-Net [2], to estimate coarse surface normals and light directions. SDPS-Net is trained with a synthetic dataset, and we used the publicly available code and model for inference[1].

For each view, SDPS-Net takes the multi-light images and the object mask as input, and estimate a surface normal map, light directions and light intensities.

1.2 More Details for Stage I

Network Architecture The network architecture of our Stage I is the same as UNISURF [9].

Training Details We use Adam as optimizer and set learning rate as 0.0001. For loss weight, we empirically adopt $\{1, 0.05, 0.005\}$ for α_{1-3}. Different from UNISURF [9], we utilized the normals estimated by SDPS-Net [2] to regularize the normals derived from the density field. The normal regularization loss was added after 1K iterations to stabilize the training. We trained Stage I for 100K iterations, which took about 12 hours to converge.

1.3 More Details for Stage II

Network Architecture We use 4-layer MLPs with width 128 for normal and albedo estimation, and a 2-layer MLPs with width 64 for predicting weights of specular SG basis. An 8-layer MLP with width 256 is used for visibility estimation. We add skip connection for normal, albedo and visibility MLPs at 2-th, 2-th and 4-th layer. We choose ReLU as the activation function.

⋆ Corresponding author
[1] https://github.com/guanyingc/SDPS-Net

Figure S1 shows the detailed network architecture of the four MLPs used in Stage II. We applied positional encoding with 10 frequency components to embed both input point $\boldsymbol{x}$ and light direction $\boldsymbol{w}_i$ into a higher dimensional space. The positional encoding is similar to [8]:

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right). \quad (1)$$

The input of the four MLPs are the encoded point location $\gamma(\boldsymbol{x})$ and encoded light direction $\gamma(\boldsymbol{w}_i)$.

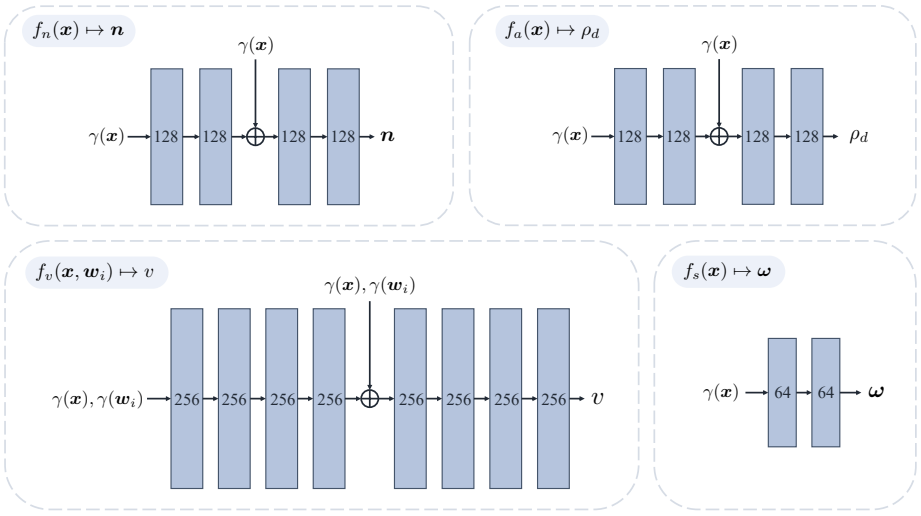


Fig. S1. Network architecture of the four MLPs in Stage II.

Training Details We use Adam as optimizer for stage II and set learning rate as 0.0005. For loss weight, we empirically adopt $\{1, 1, 1, 0.05, 0.01\}$ for β_{1-5}. To stabilize the training process, we first trained the normal MLP and visibility MLP for 5000 iterations, fixing weights of other MLPs and light parameters. We then jointly trained the normal, visibility, albedo, and specular MLPs, as well as optimize light parameters. We trained Stage II for 150K iterations, which took about 10 hours to converge.

2 More Details for the Comparison

2.1 Discussion for the Result on *COW*

Table 2 of the paper shows that our method performs slightly worse than PJ16 [10] on *COW* in the metric of Chamfer distances (i.e., 10.21 vs. 9.25). The main reason is that the camera poses in DiLiGenT-MV benchmark are located at the upper-hemisphere and slightly look downward, and the objects are placed on a desk with bottom part invisible. As a result, accurate reconstruction for the bottom part from images is impossible. Moreover, the bottom surface of *COW*) is slightly concave, which further enlarges the final mesh error.

Figure S2 visualizes the mesh reconstruction error. We can see that our method achieves more accurate reconstruction on the visible surfaces, and most of the error are in the bottom regions.

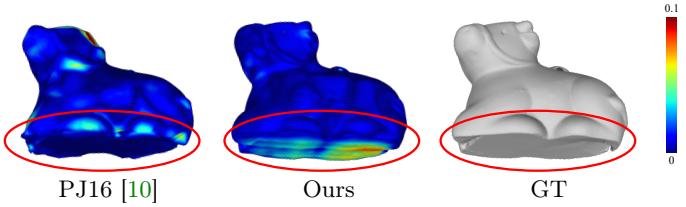


Fig. S2. Visualization of mesh reconstruction errors on *COW*. We can see that our method has larger errors on the bottom region, which is invisible in observed images and cannot be reconstructed accurately.

2.2 Discussion for the Mask Used in MAE Calculation

For fair comparison between different methods or analysis cases, we used the overlapped mask region for calculating the metrics. Therefore, there may be some small differences between the values shown in different tables.

In Table 2 of the paper, we measure the mean angular error (MAE) of the normal estimation using the intersection of input mask and our predicted mask. This is because our method reconstructs the full shape in a radiance field and uses projection to get image normals, the boundary regions might not be well aligned with the ground-truth mask. Figure S3 visualizes that when measures the normal estimation error with the ground-truth mask, our results will have a large error on the boundary region which has a thickness of about 1 pixel.

We also show the normal MAE results measured on the ground-truth mask in Table S1. Despite having larger errors on the boundary, our method still achieves the lowest average MAE on DiLiGenT-MV benchmark, which clearly verifies the effectiveness of our method.

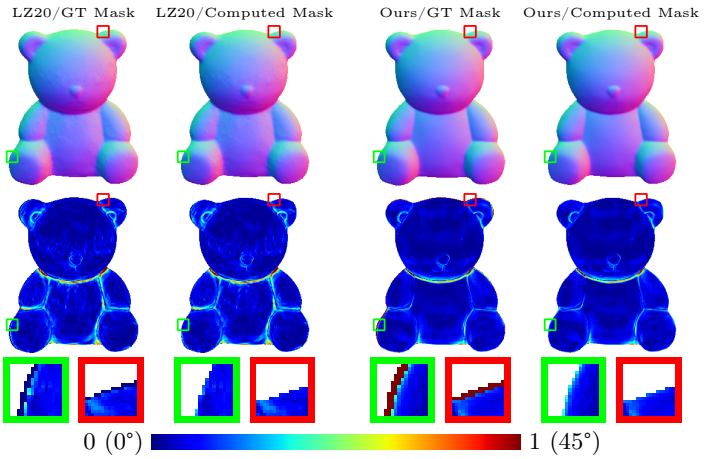


Fig. S3. Normal estimation error measured on the ground-truth mask and the computed mask of our method. As the computed mask is generate from the density field, there is a slight misalignment in the boundary region which increases the normal estimation error of our method.

Table S1. Results of normal MAE calculated with GT mask.

Method	*BEAR*	*BUDDHA*	*COW*	*POT2*	*READING*	*Average*
PJ16 [10]	12.63	14.58	13.24	15.31	12.23	13.60
LZ20 [7]	4.45	11.64	**4.13**	6.79	**8.74**	7.15
Ours	**4.39**	**11.31**	4.79	**6.29**	8.89	**7.13**

2.3 More Comparisons with Neural Rendering Methods

More Qualitative Comparisons Figure S4 and Figure S5 show the visual comparisons on two objects from DiLiGenT-MV benchmark. Our method achieves the best rendering and normal reconstruction results.

3 More Analysis for the Proposed Method

3.1 Improvement of Light Estimation

In Stage II, our method jointly optimizes the lights, normals and BRDFs. Table S2 shows the refinement of light direction and light intensity over the initialization estimated by SDPS-Net [2]. Our method significantly improves the estimation of SDPS-Net, reducing the the average MAE of light direction from 6.89 to 2.95, and average relative error of light intensity from 0.08 to 0.04.

3.2 Effect of Different Combinations of View and Light Number

To further analyze the effect of input view and light numbers, we trained our method with three different view numbers (*i.e.*, 5, 10, and 15). The camera dis-

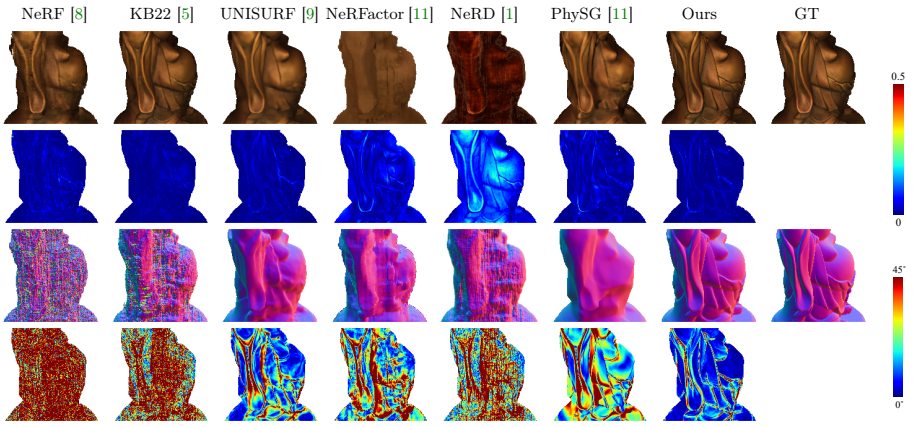


Fig. S4. More comparison with neural rendering methods on *BUDDHA* from DiLiGenT-MV benchmark.

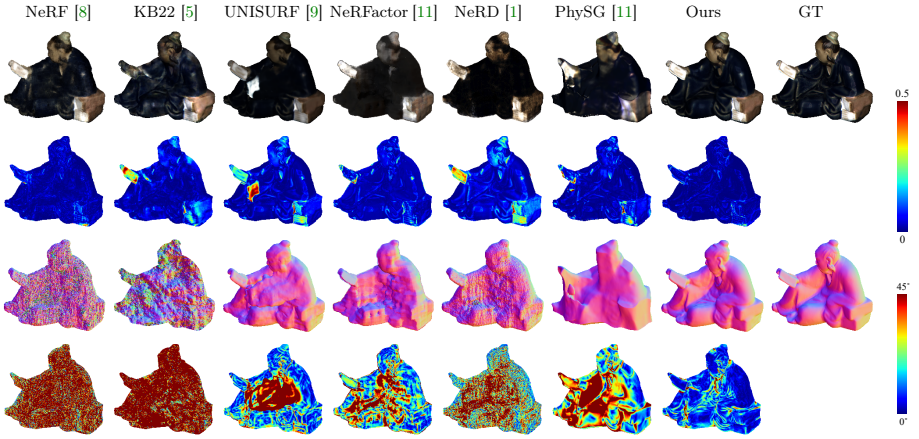


Fig. S5. More comparison with neural rendering methods on *READING* from DiLiGenT-MV benchmark.

Table S2. Improvement of light direction and intensity estimation compared to SDPS-Net [2].

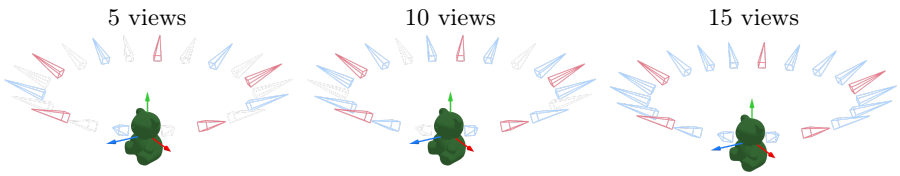
	Light Direction MAE↓						Light Intensity Error↓					
Method	BEAR	BUDDHA	COW	POT2	READING	Average	BEAR	BUDDHA	COW	POT2	READING	Average
SDPS-Net	4.90	7.17	8.55	4.73	9.09	6.89	0.10	0.06	0.11	**0.05**	0.10	0.08
Ours	**2.27**	**2.75**	**2.59**	**2.89**	**4.26**	**2.95**	**0.04**	**0.03**	**0.06**	**0.05**	**0.03**	**0.04**

tribution of the DiLiGenT-MV benchmark and the synthetic dataset are shown in Fig. S6.

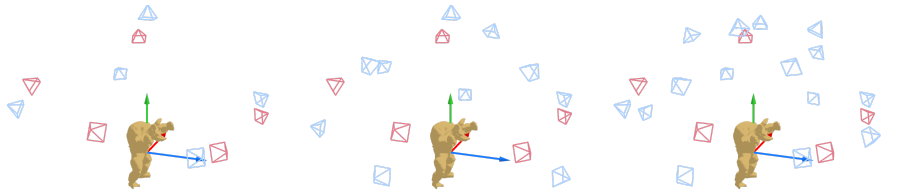
For each input view number, we experimented with four different light numbers (*i.e.*, 2, 4, 8, and 16). Note that the test camera views are the same for all experiments.

Table S3 shows the normal estimation results of our method on two challenging objects (one real object *READING* and one synthetic object *BUNNY*) with different number of views and lights. We can see that given more input views and/or light numbers can improve the shape reconstruction results, and our method can achieve robust performance using just a sparse number of views and lights.

Take results on *READING* as an example, give 5 views and 8 lights, our method achieves a MAE of 11.52, which significantly outperforms existing neural rendering methods trained with 15 views (see Table 4 of the paper, where the best performing method UNISURF [9] achieves a MAE of 19.72).



(a) Camera distribution in DiLiGenT-MV benchmark



(b) Camera distribution in the synthetic dataset

Fig. S6. Visualization of the camera distributions for setups with different input views. Training and testing views are shown in blue and red color, respectively.

Table S3. Results of our method on normal estimation error with different combinations of view and light numbers.

	READING			BUNNY		
# Lights	5 Views	10 Views	15 Views	5 Views	10 Views	15 Views
2	24.48	18.33	18.52	19.20	14.52	13.39
4	14.75	11.58	11.58	9.66	7.98	6.74
8	11.52	9.63	9.99	7.37	6.05	5.38
16	11.09	9.17	9.44	7.05	5.20	5.28

3.3 Effect of Different BRDF Parameterizations

As we found it difficult to model the specular effects of real-world objects by directly estimating the roughness parameter of the Microfacet model, we model the specular reflectance with a weighted combination of specular basis following [3,6]. Table S4 compares the results of methods using Microfacet model and specular basis on DiLiGenT-MV benchmark. We can see that the method using specular basis achieves better results in both image quality and shape reconstruction, which justifies the design of our method.

Table S4. Results of our method on DiLiGenT-MV benchmark with BRDF parameterizations.

BRDF	BEAR			BUDDHA			COW			POT2			READING			BUNNY			Average		
Modeling	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓
Microfacet	35.59	0.9836	**3.21**	28.96	0.9641	10.28	31.19	0.9754	4.21	39.45	0.9840	5.75	24.37	0.9692	9.07	22.76	0.9817	6.44	30.39	0.9763	6.49
Ours	**35.67**	**0.9837**	3.25	**29.58**	**0.9670**	**10.20**	**37.06**	**0.9890**	**4.12**	**40.01**	**0.9860**	**5.73**	**24.89**	**0.9725**	**8.87**	**25.88**	**0.9871**	**5.24**	**32.18**	**0.9809**	**6.24**

3.4 Effect of Different Material Types

To further investigate the results of our method on materials with different levels of specularity, we evaluated our method on *BUNNY* rendered with four strengths of specularity, ranging from diffuse to highly specular (denoted as A, B, C, and D). Table S5 and Fig. S7 show the results of our method on these four materials. We can see that our method achieves similar results on these four objects, indicating that our method is robust to different material types.

Table S5. Analysis on the effect of different material types on "*BUNNY*".

	Render			Normal MAE↓			Shape	Light Dir MAE↓	
Material Type	PSNR↑	SSIM↑	LPIPS↓	SDPS	Stage I	Ours	Chamfer Dist↓	SDPS	Ours
A	27.79	0.9896	0.54	10.80	8.03	5.41	4.89	9.33	1.78
B	27.61	0.9896	0.53	10.31	7.58	5.38	4.77	9.30	2.57
C	26.97	0.9888	0.55	10.53	7.91	5.55	5.03	9.33	2.49
D	26.02	0.9879	0.67	10.69	7.86	5.22	4.92	9.33	2.08

4 More Details for the Datasets

4.1 Details of the DiLiGenT-MV Benchmark

Dataset Details DiLiGenT-MV benchmark contains five objects, called *BEAR*, *BUDDHA*, *COW*, *POT2*, and *READING*. For each object, images are captured from 20 evenly distributed cameras from the same elevation (see Fig. S8 (a)) . For each view, 96 images are taken under different single directional lights with different light intensities (see Fig. S8 (b)).

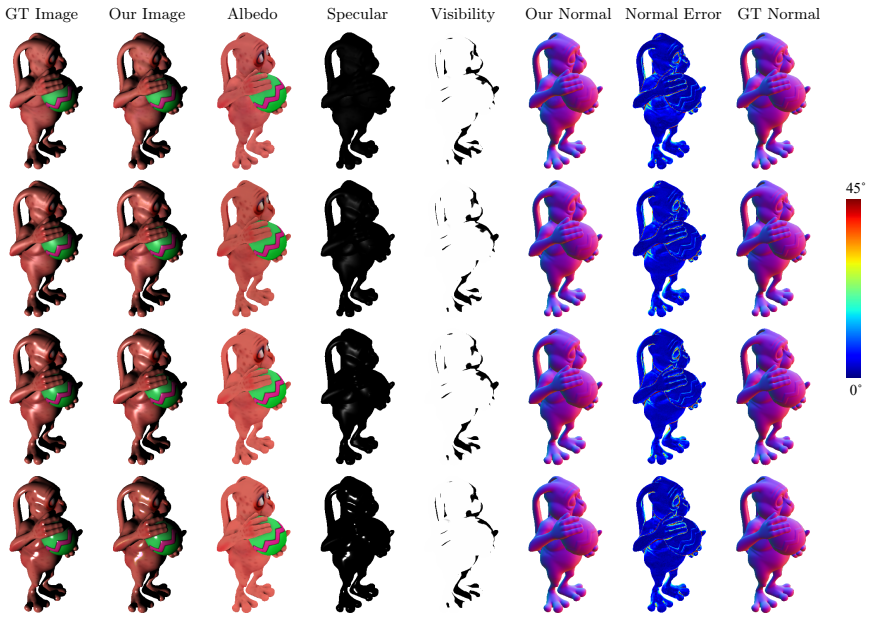
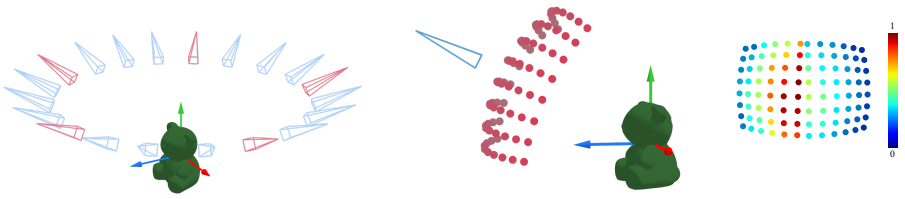


Fig. S7. Results of our method on materials with different levels of specularity. From top to bottom show the results on object A, B, C, and D. Column 1 shows the ground-truth image, and Column 2–6 show the rendered images, estimated albedo, specular component, visibility, and normals. Column 7–8 shows the normal estimation error and the ground-truth normals.



(a) Camera views (b) Distribution of light direction (left) and intensity (right)

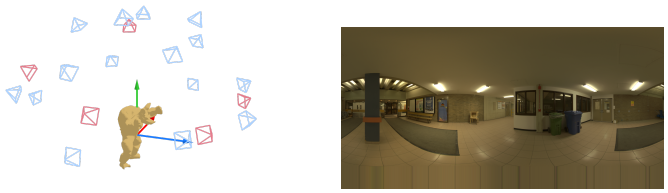
Fig. S8. Camera and light distribution in DiLiGenT-MV benchmark. The object is scaled up for 5× for easier visualization. (a) The object is located at the origin, and the cones indicate the camera poses (blue for training and red for testing views). (b) For each view, 96 images under different single light directions are captured. At the left, the position of the red point indicates the light direction. At the right, the light intensities are normalized to $[0, 1]$ and visualized with pseudo color.

Data Processing Following conventional UPS setup, we assume an unknown white directional lights setup. As we do not have access to ground-truth light intensities, we normalize the original images with light intensities predicted by [2] when training Stage I. In Stage II, we will refine light intensities during the joint optimization. To train neural rendering methods on this dataset, we normalized images according to the ground-truth light intensities.

We cropped the original images with a resolution of 612×512 into 400×400 images to remove the background for each object.

4.2 Details of the Synthetic Dataset

Rendering Details We rendered two complex objects (*i.e.*, *BUNNY*[2] and *ARMADILLO*[3]) under both PS lighting, denoted as Synth$^{\text{PS}}$ dataset, and environment lighting, denoted as Synth$^{\text{Env}}$ dataset, via Mitsuba [4]. The objects were rescaled to within a $[-1, 1]$ bounding box, and images with a resolution of 512×512 were rendered.



(a) Camera distribution (b) Environment map used for Synth$^{\text{Env}}$ dataset

Fig. S9. (a) Camera distribution used in the synthetic dataset, where 15 training views are visualized in blue color and 5 testing view are in red color. (b) An environment map used for rendering Synth$^{\text{Env}}$ dataset.

Camera & Light Distribution We used the same camera distribution for Synth$^{\text{Env}}$ dataset and Synth$^{\text{PS}}$ dataset. We randomly sampled 20 camera views on the upper hemi- sphere, where 15 views for training and 5 views for testing (see Fig. S9 (a)). For Synth$^{\text{PS}}$ dataset, we use the same light distribution as DiLiGenT-MV benchmark for each view, except that we set the same light intensity for each light. For Synth$^{\text{Env}}$ dataset, we used an indoor environment map (see Fig. S9 (b)) for both objects.

5 Applications

Our method jointly estimates surface normals, spatially-varying BRDFs, and lights. After optimization, the reconstructed object can be used for novel-view

[2] https://www.cgtrader.com/free-3d-print-models/art/sculptures/all-your-egg-are-belong-to-us

[3] http://graphics.stanford.edu/data/3Dscanrep/

rendering, relighting, and material editing. Figure S10 shows the scene decomposition, material editing, and relighting results for a novel view of objects from DiLiGenT-MV benchmark.

Please check the supplementary video for more results.

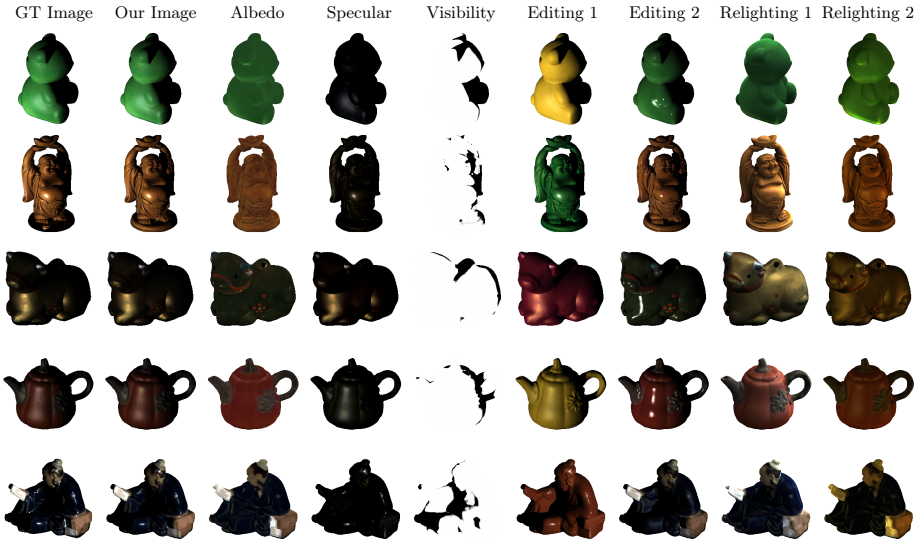


Fig. S10. Scene decomposition results of our method and applications. Columns 1–2 show the ground-truth and rendered images. Columns 3–5 show the reconstructed albedo, specular component, and visibility. Columns 6–7 edit the albedo and specular component of the objects, respectively. Columns 8–9 are two relighting results.

References

1. Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-PIL: Neural pre-integrated lighting for reflectance decomposition. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021. 5
2. Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee K. Wong. Self-calibrating deep photometric stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8739–8747, 2019. 1, 4, 5, 9
3. Zhuo Hui and Aswin C Sankaranarayanan. Shape and spatially-varying reflectance estimation from virtual exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2017. 7
4. Wenzel Jakob. Mitsuba renderer, 2010. 9
5. 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 (WACV)*, pages 1965–1977, 2022. 5

6. Junxuan Li and Hongdong Li. Neural reflectance for shape recovery with shadow handling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7
7. 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 (TIP)*, 2020. 4
8. Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421, 2020. 2, 5
9. Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5589–5599, 2021. 1, 5, 6
10. 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 (T-PAMI)*, 2016. 3, 4
11. Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5453–5462, 2021. 5