Differentiable Display Photometric Stereo

Supplemental Document

Seokjun Choi[†] Seungwoo Yoon[†] Giljoo Nam[*] Seungyong Lee[†] Seung-Hwan Baek[†]
[†] POSTECH [*] Meta Reality Labs

In this Supplemental Document, we provide additional results and details of DDPS.

Contents

1. Detailed Formulation of Photometric Stereo

We provide the detailed formulations of the normal and albedo reconstruction as follows:

$$\underbrace{\begin{bmatrix} I_1^R \\ \vdots \\ I_K^R \\ I_1^G \\ \vdots \\ I_K^G \\ I_1^B \\ \vdots \\ I_K^B \end{bmatrix}}_{\mathbf{I}} = \underbrace{\begin{bmatrix} \rho^R \\ \vdots \\ \rho^R \\ \rho^G \\ \vdots \\ \rho^G \\ \rho^B \\ \vdots \\ \rho^B \end{bmatrix}}_{\boldsymbol{\rho}} \odot \underbrace{\begin{bmatrix} \mathcal{M}_{1,1}^R & \cdots & \mathcal{M}_{1,P}^R \\ \vdots & \ddots & \vdots \\ \mathcal{M}_{K,1}^R & \cdots & \mathcal{M}_{K,P}^R \\ \mathcal{M}_{1,1}^G & \cdots & \mathcal{M}_{1,P}^G \\ \vdots & \ddots & \vdots \\ \mathcal{M}_{K,1}^G & \cdots & \mathcal{M}_{K,P}^G \\ \mathcal{M}_{1,1}^B & \cdots & \mathcal{M}_{1,P}^B \\ \vdots & \ddots & \vdots \\ \mathcal{M}_{K,1}^B & \cdots & \mathcal{M}_{K,P}^B \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} l_{1,x} & l_{1,y} & l_{1,z} \\ \vdots & \vdots & \vdots \\ l_{P,x} & l_{P,y} & l_{P,z} \end{bmatrix}}_{\mathbf{l}} \underbrace{\begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}}_{\mathbf{N}}. \tag{1}$$

$$\underbrace{\begin{bmatrix} I_1^c \\ \vdots \\ I_K^c \end{bmatrix}}_{\mathbf{I}^c} = \rho^c \underbrace{\begin{bmatrix} \mathcal{M}_{1,1}^c & \cdots & \mathcal{M}_{1,P}^c \\ \vdots & \ddots & \vdots \\ \mathcal{M}_{K,1}^c & \cdots & \mathcal{M}_{K,P}^c \end{bmatrix}}_{\mathbf{M}^c} \mathbf{lN}. \tag{2}$$

2. Details on Initial Patterns

We utilize a variety of initialization patterns, each with its own characteristics:

- **OLAT** [7]: Each OLAT pattern consists of a boundary superpixel with an intensity value of 0.9, while the other superpixels have an intensity of 0.1.
- **Group OLAT** [1]: In this pattern, we use a group of 3×3 superpixels. Each pattern activates a different group superpixel.
- **Monochromatic gradient** [5]: This pattern includes x- and y-gradient patterns in both forward and backward directions. The intensity values range from 0.1 to 0.9.
- **Monochromatic complementary** [3]: Similar to the monochromatic gradient pattern, this pattern includes x- and y-binary patterns in both forward and backward directions, with intensity values ranging from 0.1 to 0.9.
- **Trichromatic complementary** [4]: This pattern involves using complementary x-binary patterns for the red channel, complementary y-binary patterns for the blue channel, and turning on different quadrants for the green channel.
- **Trichromatic gradient** [6]: This pattern is a modification of the trichromatic complementary pattern. It includes x-gradient patterns for the red channel, y-gradient patterns for the blue channel, and a gradient from the center to the boundary for the green channel.
- **Monochromatic random**: Each superpixel intensity is randomly drawn from a uniform distribution between zero and one.
- **Trichromatic random**: Similar to the monochromatic random pattern, each superpixel intensity for each color channel is randomly drawn from a uniform distribution between zero and one.
- **Flat gray**: Each superpixel intensity is sampled from a Gaussian distribution with a mean of 0.5 and a standard deviation of 0.01.

Figure S1(a) shows the initial patterns. We set the minimum and maximum intensity values of initial patterns non-saturated from 0.1 to 0.9, to avoid zero gradient in end-to-end optimization.

3. Additional Analysis on Learned Patterns

Figure S1(c) illustrates the illumination patterns learned using DDPS with every initialization pattern. DDPS consistently improves reconstruction quality compared to initial patterns, indicating that heuristically-designed patterns can be further optimized for specific display-camera configurations. We note that the overall shape of the patterns tends to be determined during the early stages of the training process. We refer to the Supplemental Video for the progression of pattern learning.

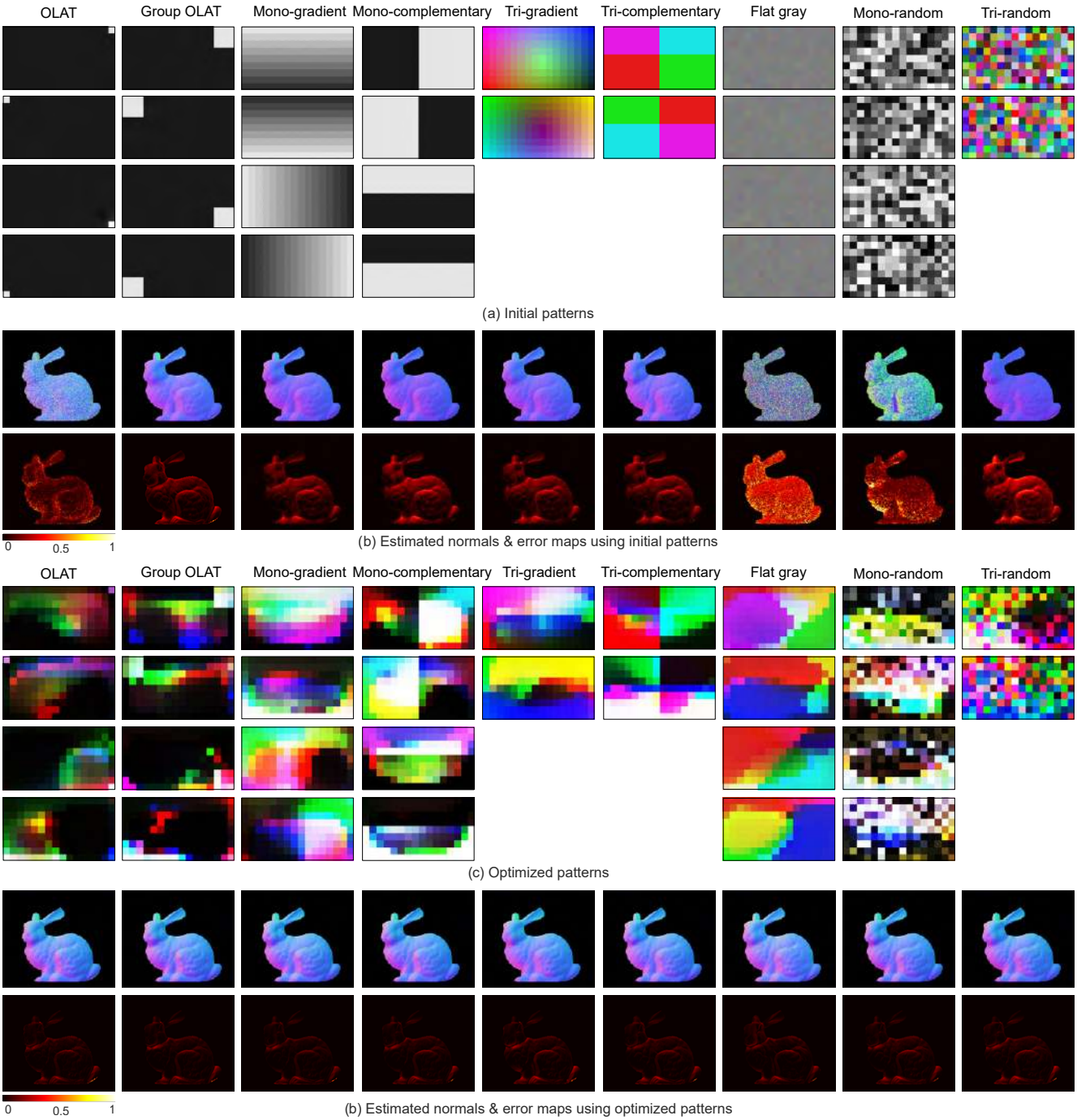


Figure S1. **Learned patterns.** (a) Heuristically-designed display patterns results in (b) sub-optimal normal reconstruction. (c) DDPS allows for learning display patterns, leading to (d) high-quality normal reconstruction.

4. Additional Analysis on the Number of Illumination Patterns

We show the impact of using varying numbers of illumination patterns for flat-gray and trichromatic-random patterns ranging from two to five. The reconstruction results on the test dataset of 3D-printed objects are presented in Table S1 computed with $\texttt{loss}(\cdot)$.

Illumination patterns	Number of patterns	Reconstruction error ↓	
		Initial	Learned
Tri-random	2	0.1461	0.0476
Tri-random	3	0.1415	0.0467
Tri-random	4	0.1096	0.0463
Tri-random	5	0.1001	0.0467
Flat gray	2	0.3549	0.0614
Flat gray	3	0.4007	0.0469
Flat gray	4	0.3930	0.0466
Flat gray	5	0.4049	0.0462

Table S1. Quantitative results of reconstructed surface normals with varying number of patterns for the trichromatic random patterns.

5. Details on Capture System

For the display, we use a commercial large curved LCD monitor (Samsung Odyssey Ark). The monitor has a 55" liquid-crystal display with 2160×3840 pixels, peak brightness of 1000 cd/m^2. Each pixel of the monitor emits horizontally linearly-polarized light at trichromatic RGB spectra due to the polarization-sensitive optical elements of LCD. We use a polarization camera (FLIR BFS-U3-51S5PC-C) with on-sensor linear polarization filters at four different angles. Thus the polarization camera captures four linearly-polarized light intensities at the angles $0°, 45°, 90°, 135°$ as $I_{0°}$, $I_{45°}$, $I_{90°}$, $I_{135°}$. Instead of using an expensive polarization camera, adopting a conventional camera with linear-polarization film is one affordable alternative. Perpendicular polarization axis of the film to the display enables capturing diffuse images.

Device Control To control the display patterns and operate the polarization camera, we use the PyGame and PySpin libraries, respectively. The devices are connected to a desktop computer via an HDMI cable and a USB3 cable. Our setup employs software synchronization between the display and the camera.

6. Iterative Normal-albedo Reconstruction

Once the surface normal $\mathbf{N}$ is obtained, we rewrite the previous Equation (2) in the main paper to solve for the albedo again:

$$\mathbf{I}^c = \rho^c \odot \mathbf{M}^c \mathbf{l} \mathbf{N}, \tag{3}$$

where $\mathbf{I}^c \in \mathbb{R}^{K\times 1}$, $\mathbf{M}^c \in \mathbb{R}^{K\times P}$ are the per-channel versions of the original vector $\mathbf{I}$ and matrix $\mathbf{M}$. For each channel $c \in \{R, G, B\}$, we estimate the albedo $\rho^c \in \mathbb{R}$ using the pseudo-inverse method as $\rho^c \leftarrow \mathbf{I}^c \left(\mathbf{M}^c \mathbf{l} \mathbf{N}\right)^\dagger$. We could iterate the normal estimation and the albedo estimation further for higher accuracy, which we found produces marginal improvements in the reconstruction quality.

We evaluate our normal-albedo reconstruction methodology iteratively on initial patterns, using the estimated albedo to calculate the subsequent normal. Our tests reveal normal-reconstruction errors of 0.0805, 0.0798, and 0.0798 for the zero-iteration, first-iteration, and second iteration respectively. These results display negligible difference, signifying that additional iterations do not significantly improve the accuracy of normal reconstruction. Consequently, for the sake of computational efficiency, we have chosen to implement a single-stage reconstruction process.

7. Optimization Details

We use a batch size of 2 and a learning rate of 0.3 with a learning-rate decay rate of 0.3 and a step size of 5 epoch. We run the training process for 30 epochs, which takes 15 minutes on a single NVIDIA GeForce RTX 4090 GPU.

8. Calibration Details

8.1. Mirror-based Geometric Calibration

We propose a mirror-based calibration method for estimating the intrinsic parameter of the camera and the location of each pixel of the monitor with respect to the camera. Figure S2(a) illustrates our geometric calibration.

We first print a checkerboard on a paper. Then, we place a planar mirror at a certain pose in front of the camera while displaying a grid of white pixels on the monitor. We then capture the mirror that reflects some of the grid points, to which the corresponding monitor pixel coordinates are manually assigned. Next, we put the printed checkerboard on top of the planar mirror and capture another image, which now contains the checkerboard. We repeat this procedure by varying poses of the planar mirror, resulting in multiple pairs of a checkerboard image and a mirror image reflecting grid points.

From the checkerboard images, we estimate the intrinsic parameter of the camera and the 3D pose of each checkerboard [8]. We then detect the 3D points of the grid points in each mirror image with the known size of the monitor and obtain the 3D points of intermediate monitor pixels via interpolation.

8.2. Radiometric Calibration

The emitted radiance from the monitor does not have a linear relationship with the pixel values of the display pattern. To account for this nonlinearity, we capture images of gray patches on a color checker under different intensity values of the display pattern. We then fit an exponential function to the captured intensity values with respect to the monitor pixel values for each color channel. Figure S2(b) shows the fitted curves.

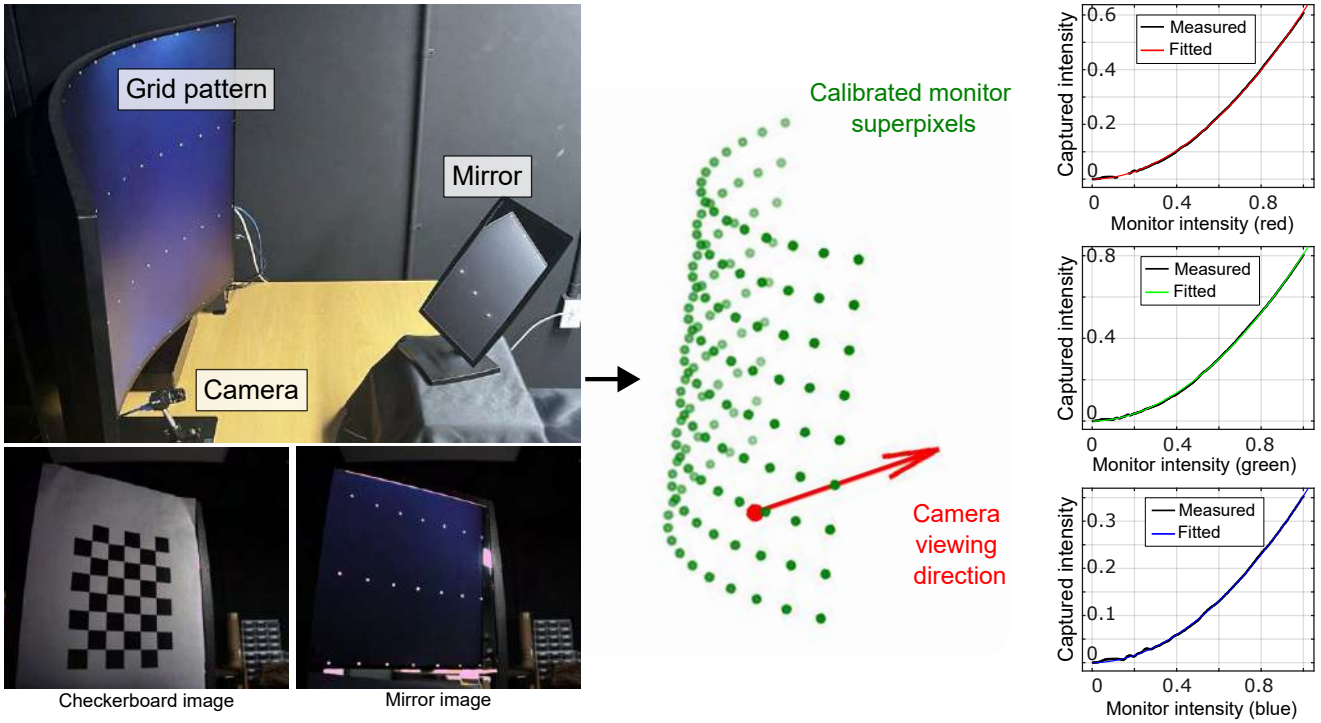


Figure S2. **Calibration.** (a) We calibrate the parameters of the camera and monitor using a mirror that reflects grid display patterns. (b) We also calibrate the non-linear mapping of monitor pixel values to emitted radiance for each color channel.

9. Example of Stokes-vector Reconstruction

To separate the diffuse and specular images, we reconstruct Stokes vector as intermediate results. Figure S3 shows the linearly-polarized images I_{0°, I_{45°, I_{90°, I_{135°, Stokes-vector elements s_0, s_1, s_2, diffuse reflection I, specular reflection S, and diffuse-specular reflection.

Figure S3. **Stokes-vector and diffuse-specular separation.** The linearly-polarized images I_{0°, I_{45°, I_{90°, I_{135°, Stokes-vector elements s_0, s_1, s_2, diffuse reflection I, specular reflection S, and diffuse-specular reflection

10. Dataset

10.1. Training and Testing Sets

Figures S8 and S7 shows the datasets used for training and testing, respectively.

10.2. 3D Printing and 3D Models

In our work, we use an FDM-based 3D printer due to its affordability and the diversity it offers. This type of 3D printer can utilize filaments of various textures, colors, and materials, thereby enhancing the diversity of our models. The 3D models are converted into gcode through a slicing process for 3D printing. It is worth noting that other types of 3D printers, such as SLA, SLS, and DLP, could further diversify the dataset. We 3D-print 11 different 3D models using a FDM-based 3D printer (Anycubic Kobra) that has a printing resolution of $\sim$0.2 mm. We use multiple filaments (PLA, PLA+, Matte PLA, eSilk-PLA, eMarble-PLA, Gradient Matte PLA, PETG) that provide diverse appearances in terms of color, scattering, and diffuse/specular ratios. The 3D-printed objects have volumes ranging from $198.9\,\mathrm{cm}^3$ to $3216.423\,\mathrm{cm}^3$. Our dataset includes 3D models, comprising busts, animal figures, and character models. These models were chosen for their diverse geometric features and asymmetry, which aids pose estimation. We foresee the potential for expanding the diversity of our models by leveraging large public 3D model datasets.

10.3. Pose Estimation and Normal Rendering

For constructing a dataset of 3D-printed objects, images are taken by the calibrated camera, under the basis illumination which is a white square on a portion of the monitor screen. With the fixed object, we took photographs of the object under a total of 144 basis illuminations, and by compositing photographs through relighting, one can synthesize a photograph of an object taken under arbitrary light sources. For ease of later pose estimation, the backgrounds of the captured images must be removed for which we adopt Adobe Photoshop for the background removal.

Even though we possess both real-world images and precise 3D model information of the objects, we need to align the object in the image with the corresponding 3D model by minimizing the reprojection error. To this end, we render silhouettes of objects using 3D mesh information and object position parameters. Then calculate the pixel-wise MSE loss between the silhouette image of the photograph and the rendered one. We used silhouettes instead of RGB rendering because it is challenging to exactly reproduce the RGB intensity given unknown reflectance. The acquired position parameters and 3D model information are used as scene parameters, and we use the normal rendering functions provided by Mitsuba3. The overall process of dataset generation is shown in Figure S5.

The pixel-wise mean squared error value is within the range of 0.0015 to 0.0028, depending on the size of the object in the image and the background removal. This low loss value signifies that the pose estimation is accurate, thereby confirming that the dataset offers a sufficiently precise representation to be deemed as ground-truth data. Figure S6 presents the qualitative results of the pose estimation accuracy. Figure S8 and Figure S7 show our training and test dataset respectively.

10.4. Sub-milimeter 3D-printing Error

Our 3D printer has 0.2mm resolution. Figure S4 shows that captured images do not present visible artifacts, which is attributed to target distance, camera FoV, and lens blur. Also, assuming a zero-mean distribution for the error, it would be canceled out as high-frequency noise during optimization.

Figure S4. **3D-printing error.** 3D-printing artifact is too small to be observed in our captured image. It can be observed in a close-up photo.

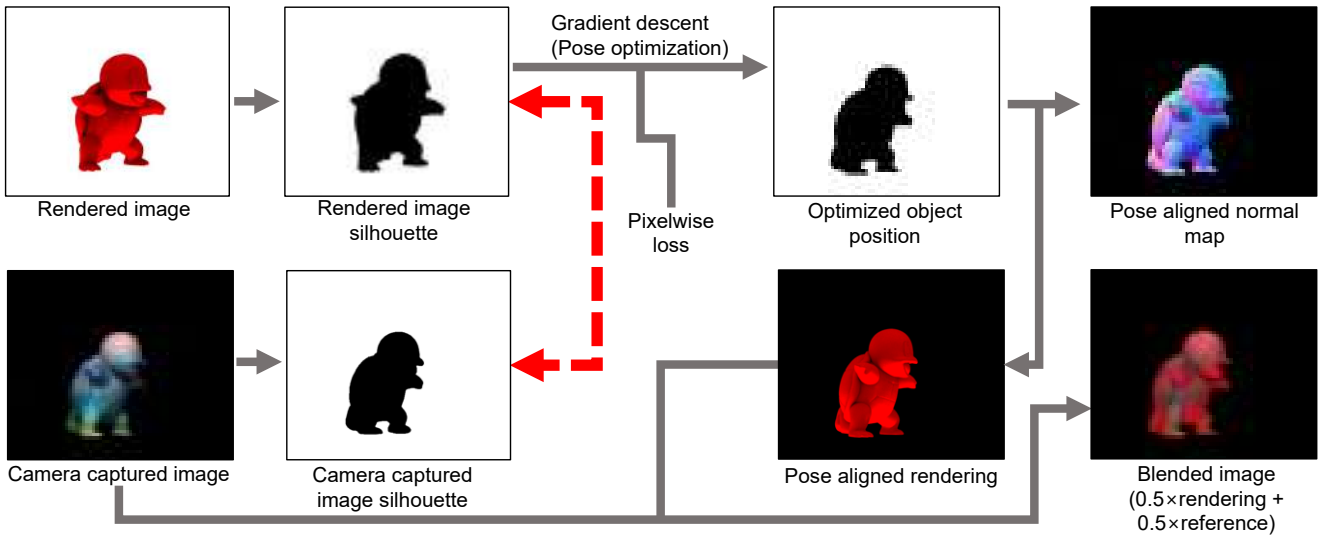


Figure S5. **The process of pose alignment.** We optimize the object position parameters using pixel-wise L2 loss between the rendered image and real-world image silhouettes. As shown in the blended image, the pose estimation process well aligns the object with the reference image, ensuring a proper correspondence between the two.

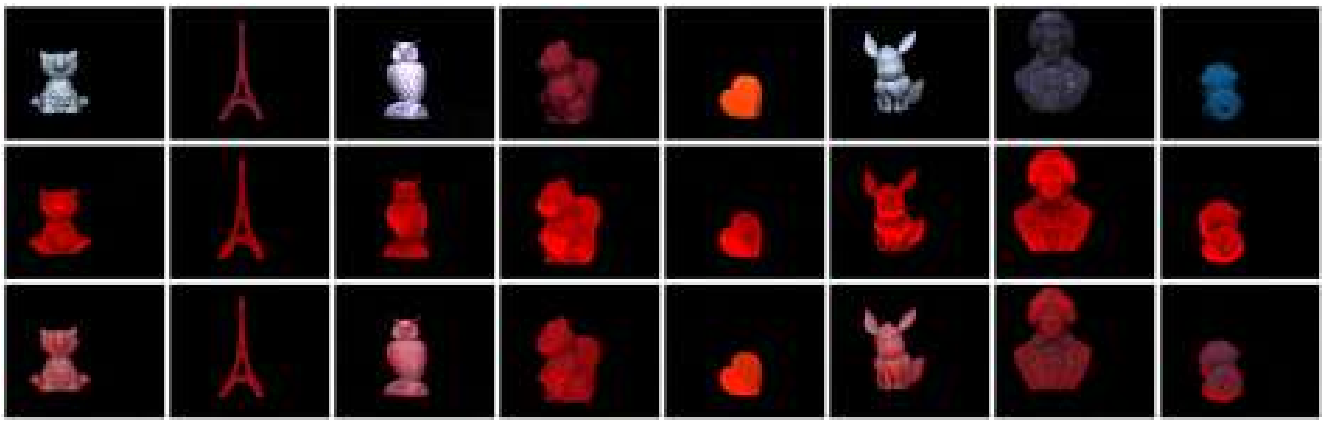


Figure S6. **A qualitative visual representation of the pose estimation results.** The first row shows captured images, the second row represents rendered images with optimized poses, and the images in the third row are blended ones which are the equally weighted sum of images in the first and second rows.

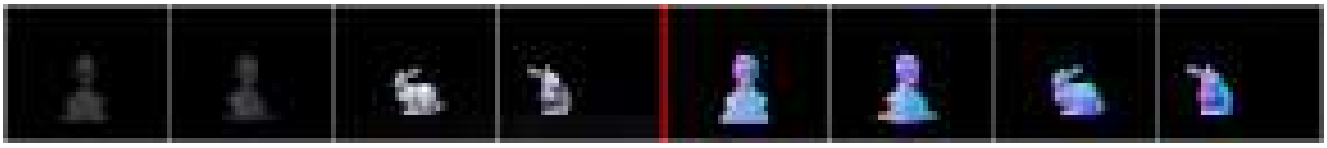


Figure S7. **Visualization of our test dataset.** Captured images are on the left and their corresponding normal maps are on the right.

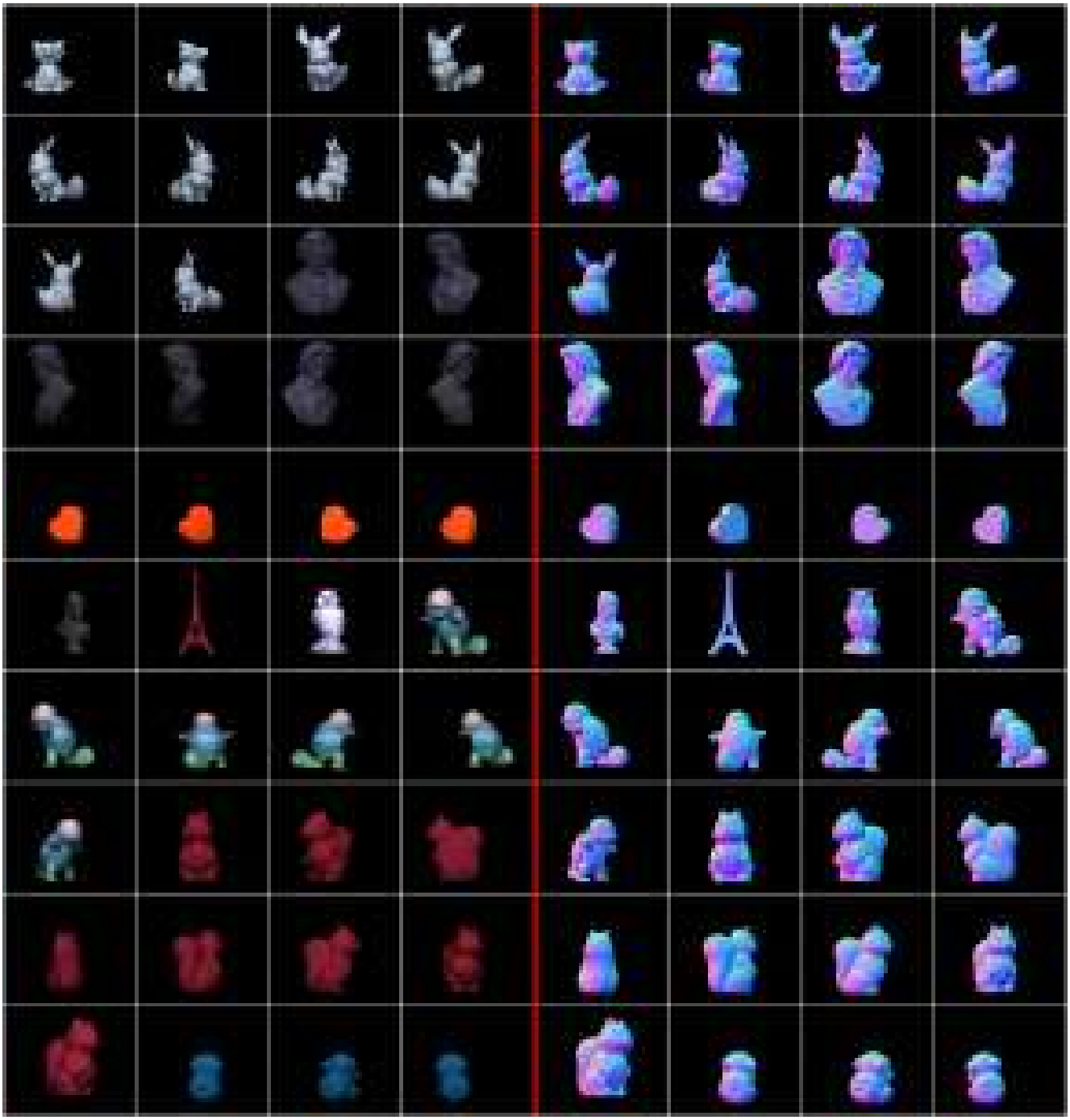


Figure S8. **Visualization of our training dataset.** Images on the left are captured images and on the right are their ground-truth normal maps.

11. Additional Discussion

11.1. Dynamic Objects

Our current experimental prototype supports software synchronization, which limits the operation speed of display-camera capture. Here, hardware synchronization would unlock the full potential of the display-camera setup by supporting the maximum framerate of the devices, reaching over 150 fps. This may require a hardware trigger mechanism between the camera, display, and GPU, which we exclude from our scope that focuses on the methodology of DDPS.

If the hardware synchronization can be implemented, we could capture polarimetric images under repeating N different monitor patterns at a framerate of 150 FPS for both display and imaging. Per each frame, we perform diffuse-specular separation and obtain diffuse image I_i for the i-th monitor pattern. This results in a duration of $1/(15N)$ seconds for capturing a scene under N patterns, which assumes marginal object movements during the capture time. Using optical flow may resolve minor alignment problem. Specifically, at any input frame, we gather $N-1$ neighboring frames, from which surface normals and diffuse albedo could be estimated by our photometric stereo method.

11.2. Superpixels

We opt to use superpixels instead of raw pixels from the display for computational efficiency. Figure S9 illustrates the similarity between the captured images when displaying a natural image using the raw display resolution and the downsampled version with superpixels. The low-frequency characteristics of projected illumination allow for using a low superpixel resolution. Although using more pixels for DDPS may enhance reconstruction accuracy by learning fine-grained patterns, the use of superpixels strikes a balance between computational efficiency and sufficient representation of the display. This is essential since GPU memory must accommodate the image formation, reconstruction, and optimization of the display patterns. Using 9×16 superpixels costs 12 GB memory. We confirmed that using 4×8 superpixels results in reconstruction error of 0.0658 comparable to 0.0443 of the 9x16 setup. We used the learned patterns initialized with four mono-gradient patterns. Exploiting native 8M display pixels leaves as an interesting future work for inverse rendering where high-frequency cue is needed.

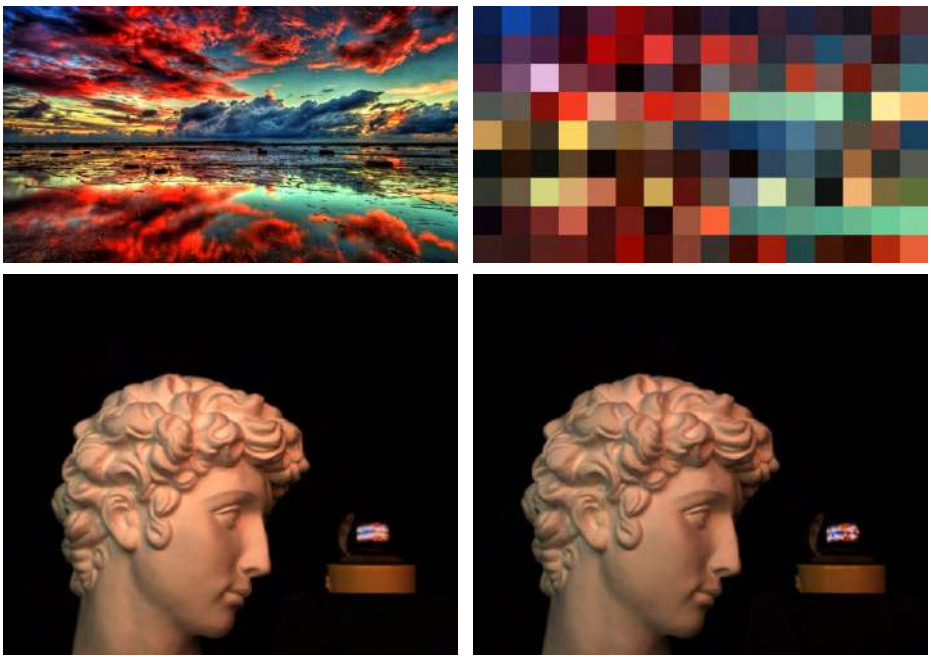


Figure S9. **Comparison on different resolution of illumination.** We compare two images under (a) the illumination of raw display resolution (2160×3840), and (b) the downsampled illumination with superpixels (9×16).

11.3. Global Illumination

Our image formation model and reconstruction method do not consider global illumination, and also our training dataset does not contain objects that incur strong global illumination. As such, DDPS fails handling objects with significant inter-reflections. We compare DDPS using initial patterns versus optimized patterns on a concave bowl. Figure S10 shows the reconstruction results and quantitative reconstruction losses with various initial/optimized patterns. The heuristic patterns (e.g., group OLAT, OLAT, monochromatic gradient) estimates more accurate normals than optimized ones. This is because the sparse initial-heuristic patterns result in less-pronounced inter-reflections. OLAT patterns show trade-off between accurate reconstruction and noisy result due to the sparsity. Some cases such as monochromatic complementary and monochromatic random patterns shows robust reconstruction on a concave bowl. This demonstrates that DDPS could potentially be improved to handle the global illumination case.

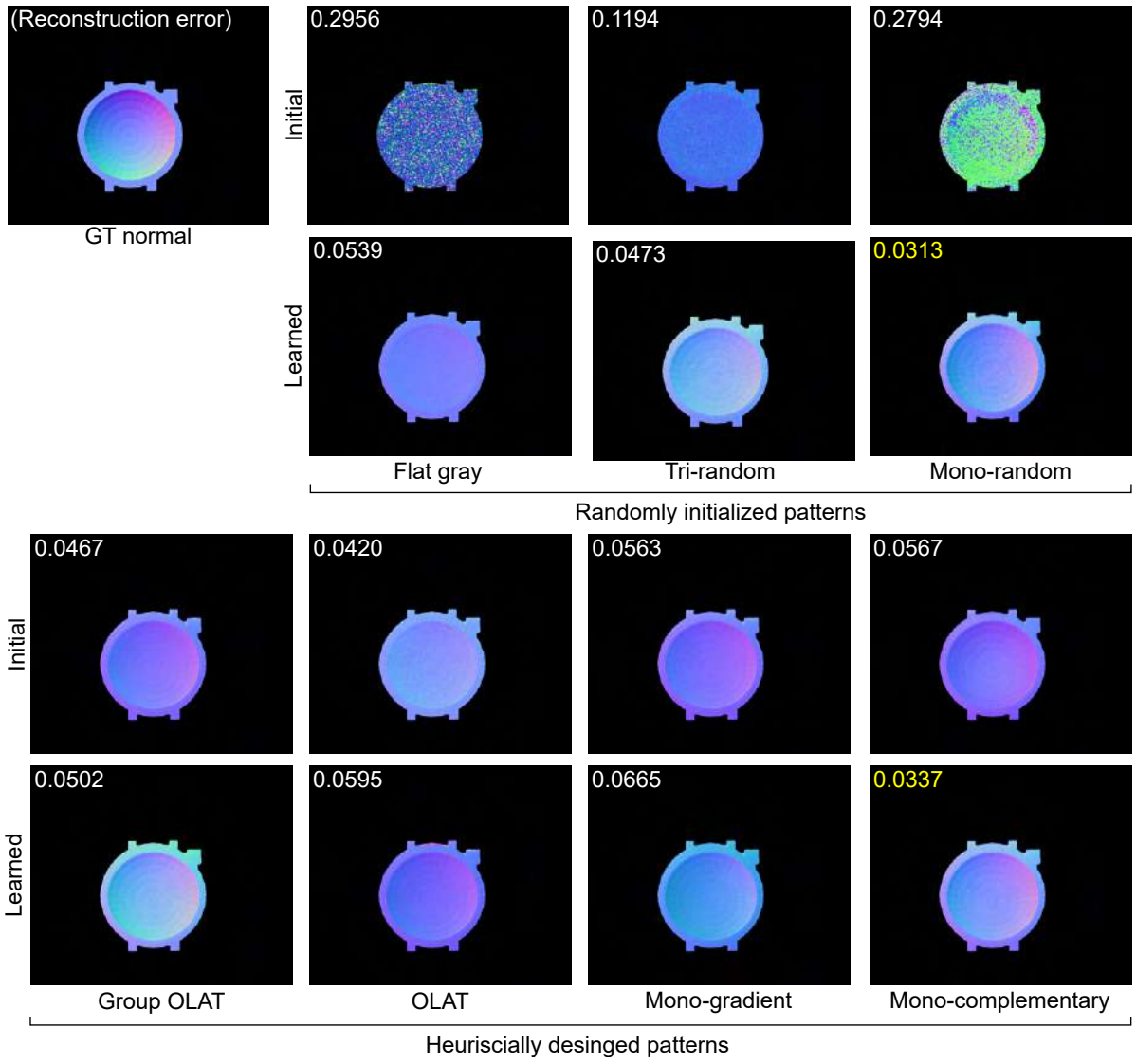


Figure S10. **Normal reconstruction on a concave bowl.** Reconstruction often fails with a concave bowl. However, some patterns like mono-random and mono-complementary shows robust reconstruction results.

11.4. Optimal Lighting versus Random Lighting

We reconstruct normals using two different 65-frame set sampled from the *Big Buck Bunny* video where the intervals within frame are 0.3/9 seconds respectively. Figure S11 shows frames, reconstructed normals, and error maps. The inter-frame standard deviations of each frame set are 0.2094/0.2658 respectively. We use 65 frames, which can be stored in our GPU memory limit. The reconstruction errors are 0.1893/0.1140 for the two videos, showing the dependency of video contents on reconstruction accuracy. In contrast, using learned patterns with DDPS achieves the reconstruction error of 0.0476 with only two patterns (Tri-random, Table S1). Dynamic photometric stereo with two learned patterns would be an interesting future work.

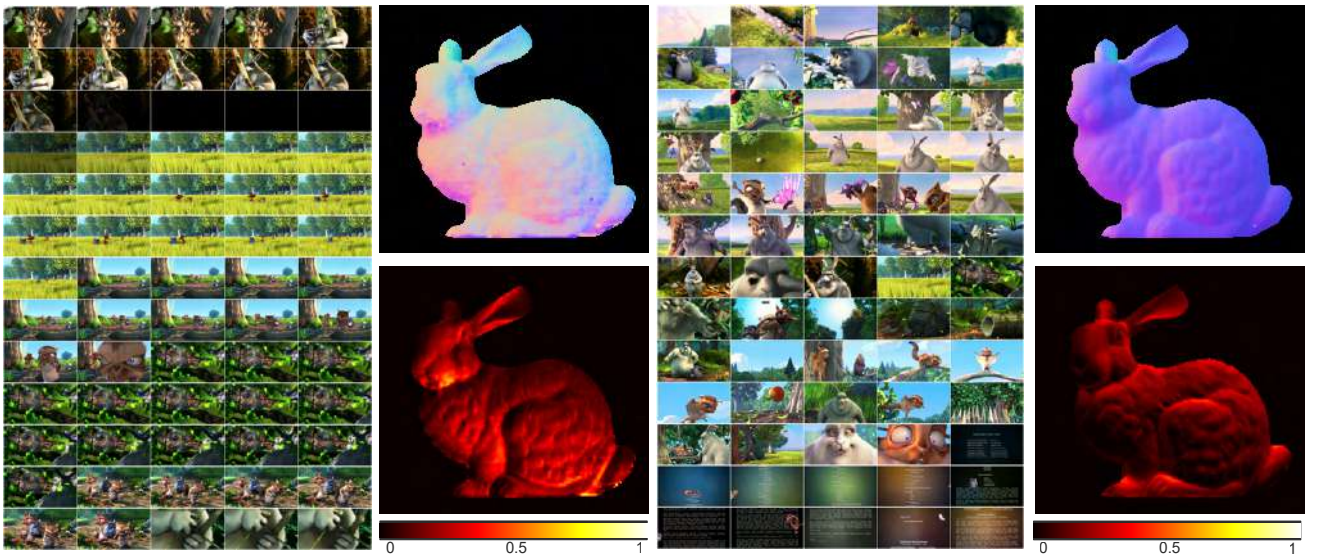


Figure S11. **Reconstruction with video frames.** (a) Video frame set with 0.3 second intervals shows (b) estimated normals (top) and error maps (bottom). (c) Video frame set with 9 second intervals shows (d) estimated normals (top) and error maps (bottom).

11.5. Generalizability of Learned Patterns

We conduct cross validation for demonstrating the generalizability of DDPS. Table **??** shows that DDPS achieves consistently low reconstruction errors and similar characteristics of learned patterns for the cross-validation test.

(Mean/std. dev.) of recon. error	(Mean/std. dev.) of learned-pattern intensity
(0.0457/0.0054)	(0.4371/0.0842)

Table S2. **Statistics of reconstruction error and learned patterns.** 5-fold cross validation shows consistent reconstruction error and learned-pattern intensity with low std. dev..

11.6. Scene Geometry Assumption

Due to inaccessibility of accurate geometry of the inference scene, we assume that the surface points of objects lie on a plane located 50 cm away from the camera position along the z-axis. This assumption is critical for normal estimation in conventional methods as it interrupt utilizing ground-truth lighting vectors. To demonstrate the robustness of DDPS under such assumption, we conduct a comparison experiment between patterns learned using ground-truth lighting vectors from ground-truth depth and patterns learned using proposed method. In the former case, it shows reconstruction error of 0.0467 with using GT depth. In comparison, DDPS achieves reconstruction error 0.0475 without using GT depth. These results implicate the robustness of DDPS in learning patterns that can compensate deviations in scene geometry assumptions.

11.7. Display Size

We test DDPS on a simulated 32” display by sampling 5×10 central superpixels of the original 55” display. Normal reconstruction learned and tested on the 32” display gives the reconstruction error of 0.0659, when using mono-gradient initialization. Even though this error is larger than that of using the original 55” display (0.0443), the learned patterns enables outperforming the best reconstruction accuracy of 0.0805 on a 55” display using heuristic patterns, group OLAT.

11.8. Generalizability to Arbitrary In-the-wild Shapes

Figure S12 shows that learned patterns improves normal reconstruction for in-the-wild objects.

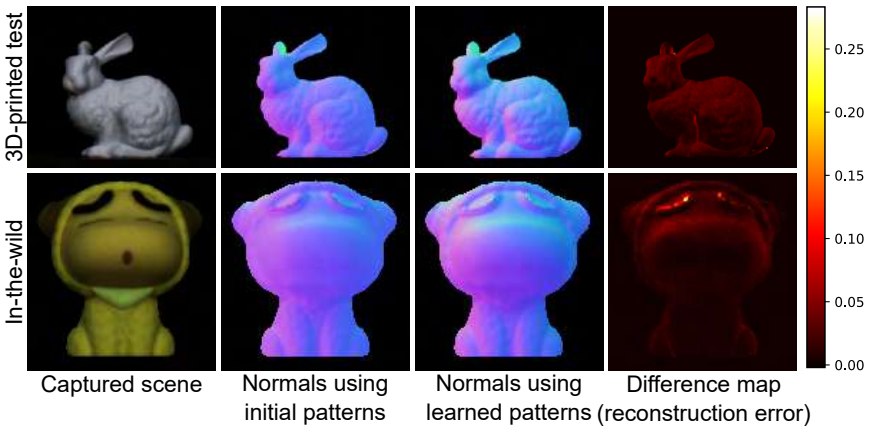


Figure S12. **DDPS for test/in-the-wild objects.**

11.9. Comparison with Learning-based Photometric Stereo

We compare the reconstruction results with DDPS to state-of-the-art normal reconstruction method, SDM-UniPS [2] that leverages neural networks. SDM-UniPS uses mask-free inputs and supports unknown and arbitrary lighting conditions. Figure S13 show the reconstruction results of SDM-UniPS on our test dataset. The uncalibrated learning-based methods are often fragile with complex lighting contexts or out-of-distribution objects, and also cannot fully leverage benefits of carefully-designed illumination. In contrast, the physically-valid reconstruction methods enables DDPS robust on aforementioned cases.

12. Additional Results

We show additional results of DDPS in Figures S15 and S16, including captured images, their respective illumination patterns, surface normals, and diffuse albedo. We also provide a failure example due to strong highlights.

12.1. Results with Different Learned Patterns

Figure S14 shows that reconstruction results with different learned patterns are generally similar. Section 11.3 further shows that severe inter-reflection in a concave bowl makes notable difference between learned-pattern results.

12.2. Diffuse Albedo

Figure S15 and S16 shows the reconstructed surface normals and diffuse albedo of various objects including a human face from four input images captured using the learned patterns.

12.3. Robustness against Ambient Illumination

We experimentally demonstrate testing our learned patterns while ambient light is present. To this end, we capture an additional image under a black display pattern to capture the contribution only from ambient light. We then subtract this ambient-only image from the images taken under the learned display patterns with ambient light. This enables isolating the display-illuminated components only. We then use photometric-stereo reconstruction for obtaining surface normals. To handle the limited dynamic range of the display and the camera, we use HDR imaging for obtaining high-quality normal reconstruction. FigureS16 shows the reconstructed surface normals.

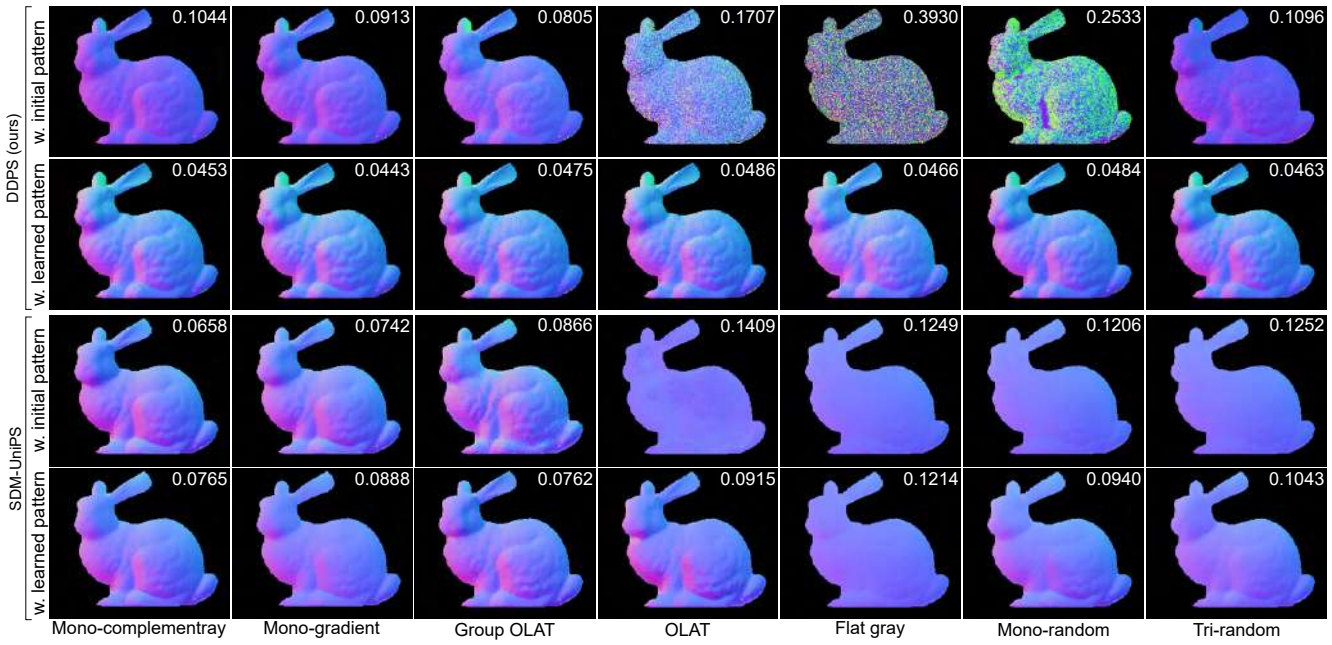


Figure S13. **Comparison with SDM-UniPS.** The reconstruction error is indicated in the upper right corner of the each reconstructed normal. The uncalibrated learning-based methods often fails on leveraging lighting context.

Figure S14. **Results with different learned patterns (top vs. bottom).** DDPS shows similar qualitative reconstruction results with different learned patterns.

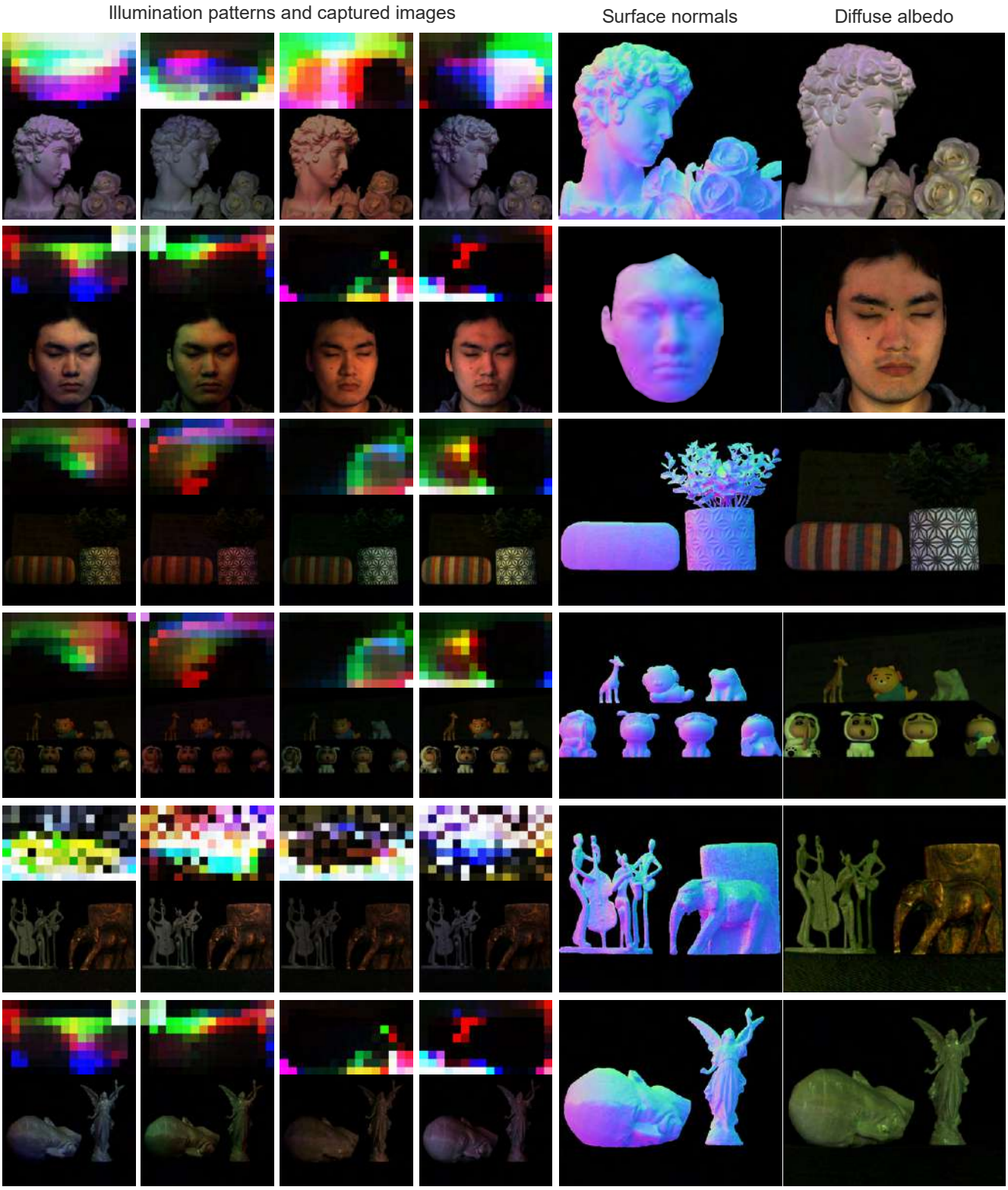
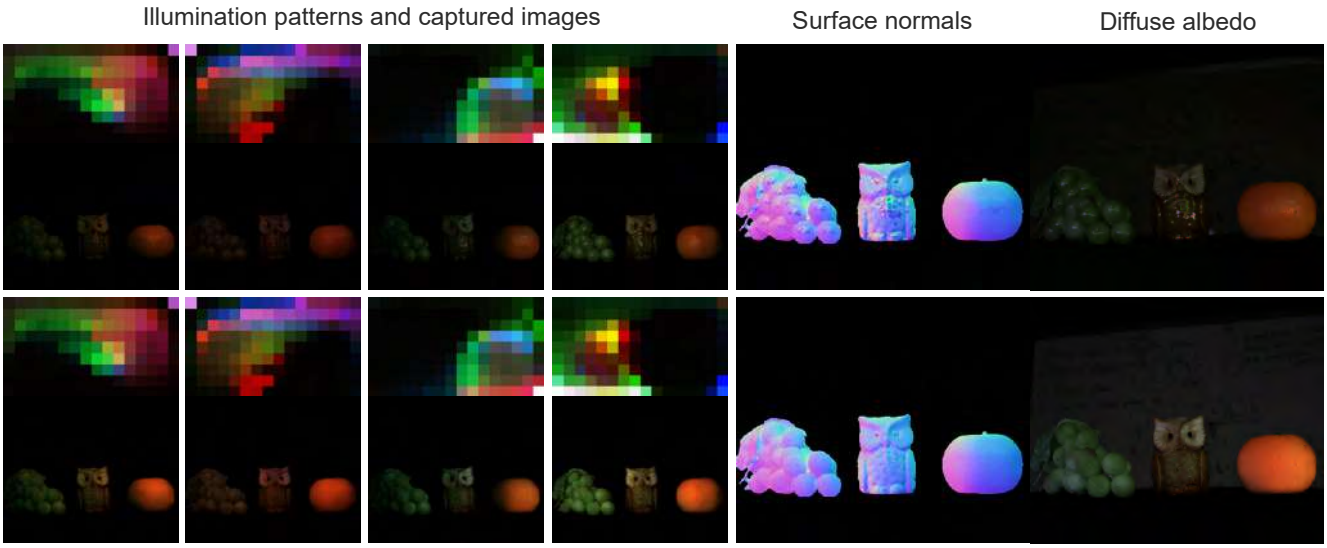


Figure S15. **Additional results of DDPS.**



(a) Diffuse + specular inputs, outputs (first row) and Diffuse inputs and outputs (second row)

(b) Ambient inputs, outputs (first row), an image under a black display pattern (second row), and the monitor-illuminated components (third row)

(c) Failure case

Figure S16. **Additional results of DDPS.**

References

[1] Sai Bi, Stephen Lombardi, Shunsuke Saito, Tomas Simon, Shih-En Wei, Kevyn Mcphail, Ravi Ramamoorthi, Yaser Sheikh, and Jason Saragih. Deep relightable appearance models for animatable faces. *ACM Trans. Graph.*, 40(4):1–15, 2021. 3

[2] Satoshi Ikehata. Scalable, detailed and mask-free universal photometric stereo. *arXiv preprint arXiv:2303.15724*, 2023. 14

[3] Christos Kampouris, Stefanos Zafeiriou, and Abhijeet Ghosh. Diffuse-specular separation using binary spherical gradient illumination. In *EGSR (EI&I)*, pages 1–10, 2018. 3

[4] Alexandros Lattas, Yiming Lin, Jayanth Kannan, Ekin Ozturk, Luca Filipi, Giuseppe Claudio Guarnera, Gaurav Chawla, and Abhijeet Ghosh. Practical and scalable desktop-based high-quality facial capture. In *Eur. Conf. Comput. Vis.*, pages 522–537. Springer, 2022. 3

[5] Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Eur. Conf. Render. Tech.*, pages 183–194, 2007. 3

[6] Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhöfer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, et al. Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. *ACM Trans. Graph.*, 38(4):1–12, 2019. 3

[7] Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T Barron, and Ravi Ramamoorthi. Light stage super-resolution: continuous high-frequency relighting. *ACM Trans. Graph.*, 39(6):1–12, 2020. 3

[8] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. 6