nLMVS-Net: Deep Non-Lambertian Multi-View Stereo -Supplementary Material-

Kohei Yamashita Yuto Enyo Shohei Nobuhara Ko Nishino
Graduate School of Informatics, Kyoto University, Kyoto, Japan
https://vision.ist.i.kyoto-u.ac.jp/

1. Reflectance Estimation

Figure 1 depicts how we update reflectance parameters using the geometry estimates and the input view images. We optimize the reflectance parameters so that errors between the input view images and the rendered images for each view are minimized. Using the current geometry estimates, *i.e.*, the estimated depths and the surface normals for a reference view, we can synthesize the rendered images for each view by computing irradiance (Eq. (1) of the main paper) of each surface point and projecting it onto the image plane. The projection onto the neighboring views, however, makes the rendered images sparse. Instead, we warp both the input and the rendered images into the reference view so that the errors can be computed densely for each pixel in the reference view. We warp the input view images $I_k(\boldsymbol{m})$ into the reference view as

$$\log I'_k(\boldsymbol{m}) = \sum_d \hat{p}(d;\boldsymbol{m}) \log I_k(\boldsymbol{m_k}(\boldsymbol{m},d)), \qquad (1)$$

where $\boldsymbol{m_k}(\boldsymbol{m},d)$ is the mapping from the pixel coordinate and depth of the reference view into the pixel coordinate in the input view. We use the depth probability volume $\hat{p}(d;\boldsymbol{m})$ instead of the final depth estimate $\hat{d}(\boldsymbol{m})$ so that we can evaluate the consistency even if the depth probability volume is multimodal. Similarly, we synthesize a warped version of the rendered image as

$$\log E'_k(\boldsymbol{m}) = \sum_d \hat{p}(d;\boldsymbol{m}) \log E(\boldsymbol{v_k}, \hat{\boldsymbol{n}}(d,\boldsymbol{m})), \qquad (2)$$

where $E(\boldsymbol{v},\boldsymbol{n})$ is the radiometric image formation model (Eq. (1) of the main paper), $\boldsymbol{v_k}$ is the viewing direction of the input view, and $\hat{\boldsymbol{n}}(d,\boldsymbol{m})$ is the surface normal estimate sampled from the normal volume.

A straightforward approach to compute errors between these images would be pixel-wise intensity errors. Such naive error computation, however, is too brittle as these images are noisy due to the reconstruction errors of the depths and surface normals. We instead Gaussian blur the (warped) input view images and the (warped) rendered images and then compute the pixel-wise errors as

$$L_{\mathrm{c}}(\boldsymbol{m}) = \sum_k \left\| \overline{\log I'_k}(\boldsymbol{m}) - \overline{\log E'_k}(\boldsymbol{m}) \right\|_1 , \qquad (3)$$

where $\overline{\log I'_k}(\boldsymbol{m})$ is the blurred input view image and $\overline{\log E'_k}(\boldsymbol{m})$ is the blurred rendered image.

Although Eq. (3) ensures consistency between the estimated reflectance and geometry, appearance details, such as sharp specular highlights, can still be missed. For this, we derive another metric to evaluate the consistency at a finer level. The key idea is that the ground-truth surface normal should exist around the estimated one which almost exactly satisfies the radiometric consistency. For each pixel $\boldsymbol{m}$ and each discretized depth hypothesis d, *i.e.*, for each candidate of the surface points, we find such orientation $\tilde{\boldsymbol{n}}(d,\boldsymbol{m})$, as one that locally minimizes

$$\sum_k \left\| \log I_k(\boldsymbol{m_k}(\boldsymbol{m},d)) - \log E\left(\boldsymbol{v_k}, \tilde{\boldsymbol{n}}(d,\boldsymbol{m})\right) \right\|_1 . \qquad (4)$$

By using this refined surface normal estimate $\tilde{\boldsymbol{n}}(d,\boldsymbol{m})$, we synthesize a refined version of the (warped) rendered image as

$$\log \tilde{E_k}'(\boldsymbol{m}) = \sum_d \hat{p}(d;\boldsymbol{m}) \log E(\boldsymbol{v_k}, \tilde{\boldsymbol{n}}(d,\boldsymbol{m})) . \qquad (5)$$

We evaluate the finer-level consistency as pixel-wise errors between this image and the warped image

$$L_{\mathrm{f}}(\boldsymbol{m}) = \sum_k \left\| \log I'_k(\boldsymbol{m}) - \log \tilde{E'_k}(\boldsymbol{m}) \right\|_1 , \qquad (6)$$

and use the sum of Eq. (3) and Eq. (6) as the loss functions of reflectance estimation.

1.1. Occlusion Detection

Another challenge of the reflectance estimation is that the loss functions may suffer from occlusions. Although we empirically found that the proposed loss functions are

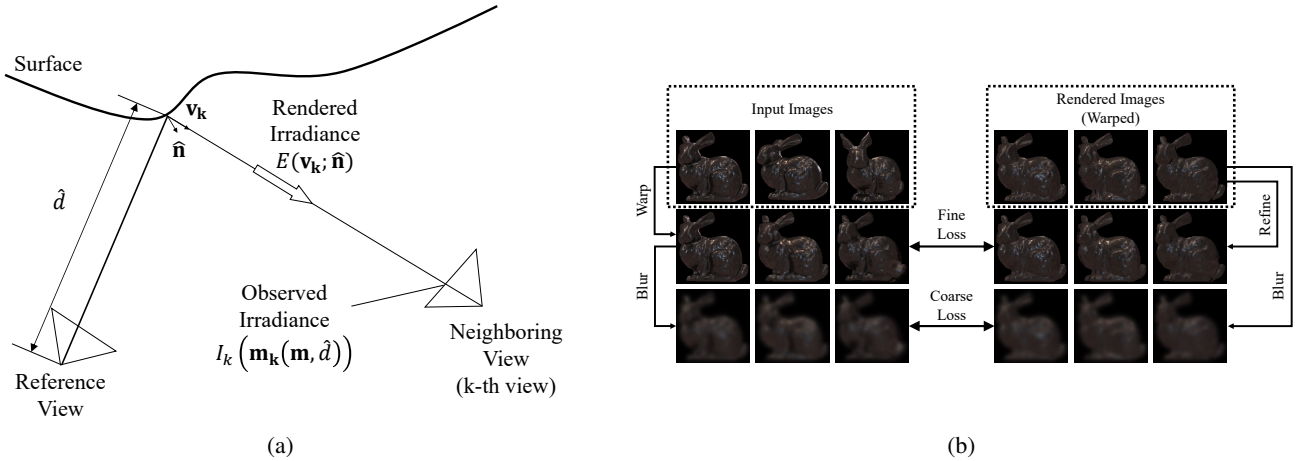


Figure 1: (a) We update the reflectance parameters by minimizing errors between pixel values of the input view images $I_k(\boldsymbol{m_k}(\boldsymbol{m}, \hat{d}))$ and those rendered using the current surface normal estimates, *i.e.*, $E(\boldsymbol{v_k}, \hat{\boldsymbol{n}})$. (b) We can compute this error for each pixel in the reference view by warping both the input and the rendered images into the reference view. As straightforward pixel-wise intensity errors are too brittle due to reconstruction errors of geometry, we instead derive two loss functions. First, we Gaussian blur both images and then compute pixel-wise errors. This ensures consistency at a coarse level. Second, we find a (pseudo) ground-truth surface normal around the estimated one that almost exactly satisfies the radiometric consistency and use it to refine the rendered image. This ensures consistency at a fine level, *e.g.*, consistency of sharp specular highlights.

robust enough to such deviations even if we simply use two neighboring views for the error computation, we can also explicitly handle occlusions when we recover the whole 3D shape (Sec. 3.4 of the main paper), *i.e.*, when we have depth estimates for each input view. For this, inspired by reprojection loss of self-supervised binocular and multi-view stereo, we compute visibility of the reference view's pixels from neighboring views using the depth estimates of these views. Figure 2 depicts how we evaluate the visibility. We compute a visibility score based on the error between the depth in the k-th neighboring view ($\bar{d}_k$) computed from the reference view's depth estimate ($\hat{d}_0$) and that sampled from the neighboring view's estimates ($\hat{d}_k$). As $\bar{d}_k$ and $\hat{d}_k$ are consistent only if the surface point is visible from the neighboring view, we compute the visibility score s_k as

$$s_k = \exp(-|\bar{d}_k - \hat{d}_k|)\,, \tag{7}$$

and use two neighboring views that maximize this score for the error computation.

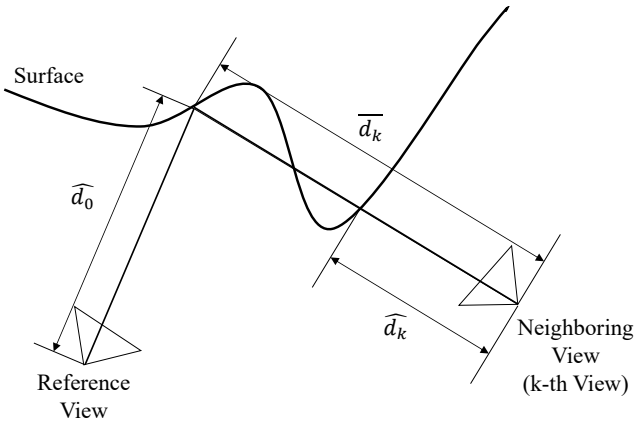


Figure 2: When we recover the whole 3D shape from 10 (or more) multi-view images (Sec. 3.4 of the main paper), *i.e.*, when we have depth estimates for each input view, we compute approximate visibility of a pixel in the reference view from neighboring views. We compute the visibility based on the error between the depth in the neighboring view ($\bar{d}_k$) computed from the reference view's depth estimate ($\hat{d}_0$) and that sampled from the neighboring view's estimates ($\hat{d}_k$).

2. Test set of nLMVS-Synth Dataset

As shown in Fig. 3, the test set of nLMVS-Synth dataset consists of 4320 multi-view images of 216 different combinations of 6 shapes [10, 14, 1], 6 materials [11], and 6 illuminations [5, 9].

3. Image Capture Flaws in Multiview Objects Under Natural Illumination Database [12]

As shown in Fig. 4 and briefly described in the main paper, the Multiview Objects Under Natural Illumination

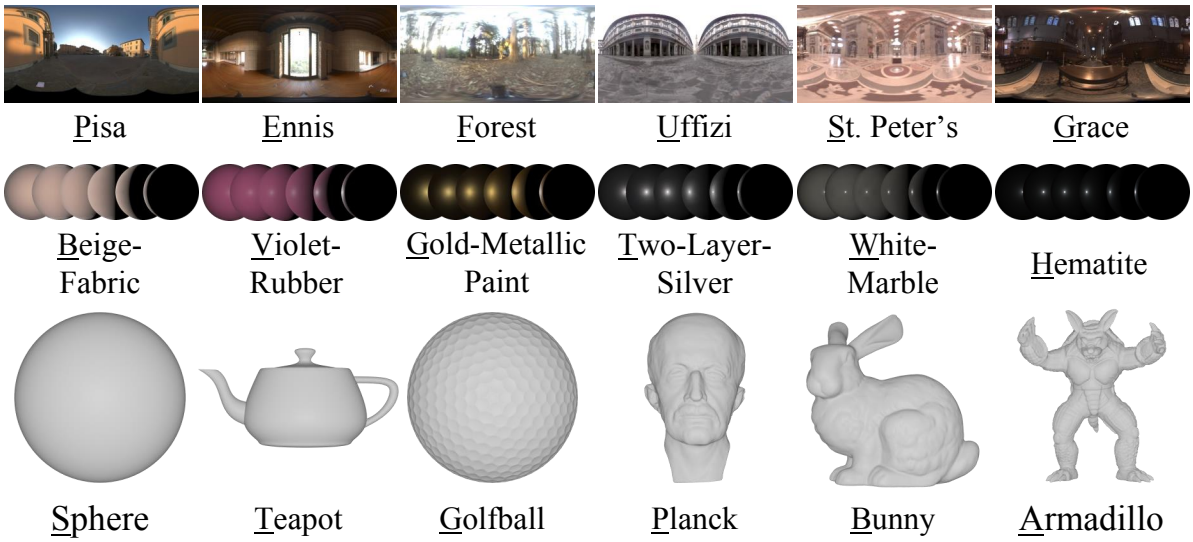
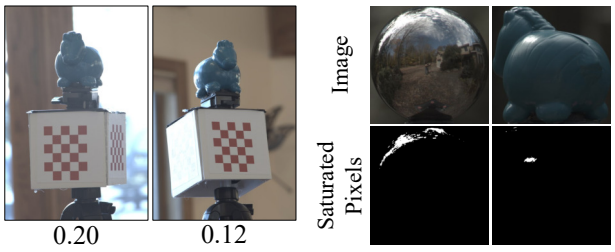


Figure 3: The test set of our novel synthetic image dataset, nLMVS-Synth, consists of 4320 rendered images of 216 combinations of 6 shapes [10, 14, 1], 6 materials [11], and 6 illumination environments [5, 9].



(a) Glare (b) Saturation

(c) Poor calibration

Figure 4: Image capture flaws in Multiview Objects Under Natural Illumination database [12] including lens glare, large saturations, and poor geometric calibration. (a) The first image is much brighter than those of the neighboring views due to lens glare. The numbers are median brightness of the object. (b) RAW Images taken with the shortest exposure time (provided by the authors of [12]). Specular highlights are completely saturated. (c) We overlay the silhouette of the ground-truth 3D mesh model on an input image. Due to poor geometric calibration, the ground-truth 3D mesh is not consistent with the input image.

Database [12] contains clear flaws in image capture including saturation, glare, and poor geometric calibration. For these reasons, although we run our method on this dataset for qualitative comparison with Oxholm and Nishino [13], experimental results on this dataset do not accurately reflect superiority of any method.

4. Implementation Details

4.1. Network Architecture

Figure 5 and Tab. 1 show the architecture of the shape-from-shading network. It consists of standard fully connected and 2D convolutional layers. As described in the main paper, we use this network recursively (3 times) to refine the observed surface normal likelihood.

Figure 6 and Tab. 2 show details of "Pixel-Wise Encoding," "3D Conv," and "3D UNet" inside the cost volume filtering network (Fig. 4 of the main paper). The architecture of "2D UNet" is the same as "ImageFeaNet" inside the shape-from-shading network except the numbers of input and output channels. The numbers of the input and output channels of "2D UNet" are 22 and 32, respectively.

4.2. Training

We train our networks on the training set of the nLMVS-Synth dataset. As described in the main paper, we train the shape-from-shading network and the cost volume filtering network separately. We use the Adam optimizer with default parameters of PyTorch 1.8.1 for both training.

For training of the shape-from-shading network, we set the batch size and the number of epochs as 2 and 40, respectively. We run the training on a single NVIDIA RTX A6000

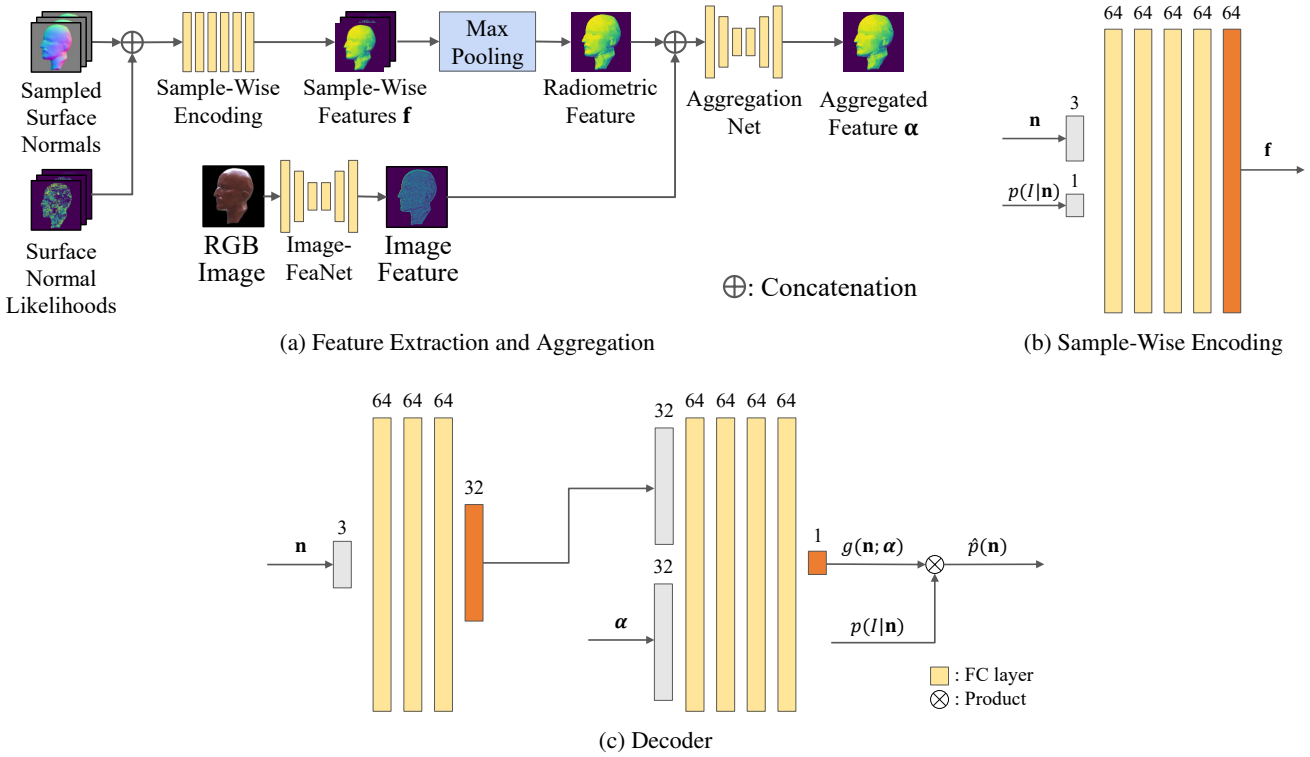


Figure 5: Architecture of the shape-from-shading network. Please also refer to Tab. 1 for details of the "ImageFeaNet" and the "AggregationNet". (a) We extract and aggregate features from an input RGB image, sampled surface normals, and their observation likelihoods. (b) We use an MLP to extract features from a sampled surface normal $\boldsymbol{n}$ and its observation likelihood $p(I|\boldsymbol{n})$. Each fully connected (FC) layer except the final layer is followed by a leaky ReLU activation function. (c) Using an aggregated feature $\boldsymbol{\alpha}$ and a decoder MLP, we compute the output (unnormalized) probability density distribution $\hat{p}(\boldsymbol{n})$. In the final layer of the MLP, we use a SoftPlus activation so that the output $g(\boldsymbol{n};\boldsymbol{\alpha})$ is positive. For the other layers, we use leaky ReLU activation.

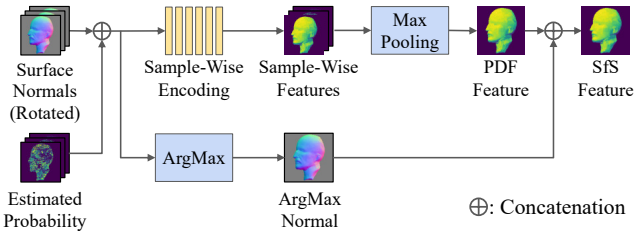


Figure 6: Details of "Pixel-Wise Encoding" inside the cost volume filtering network. The "Sample-Wise Encoding" is similar to the one in the shape-from-shading network (Fig. 5b), but the number of hidden layers is 3 and the output is 16-dimensional. For each pixel, the "ArgMax" outputs a surface normal orientation that maximizes the estimated probability. We empirically found that the combination of the "PDF" (probability density function) feature and the "ArgMax Normal" is effective to recover accurate geometry.

GPU (48GB). It takes approximately 100 hours.

For training of the cost volume filtering network, we set the batch size and the number of epochs as 4 and 30, respectively. We run the training on two NVIDIA RTX A6000 GPUs (48GB×2). It takes approximately 100 hours. We use images of three views (a reference view and its two neighboring views) as inputs of the network during this training. Please note that, at inference time, the network can take an arbitrary number of images as inputs. At inference time (*i.e.*, for the joint shape and reflectance estimation), we use five view images to construct a cost volume.

4.3. Joint Shape and Reflectance Estimation

We start the alternating estimation of the geometry and reflectance from an initial estimate of the reflectance. As the conditional iBRDF model is trained so that its latent vector is always within the unit hypersphere, we use a zero vector as an initial estimate of the latent vector $\boldsymbol{z}$. We initialize the channel-wise scale factor $\boldsymbol{c}$ so that the mean irradiance $E(\boldsymbol{\omega_o}, \boldsymbol{n})$ (Eq. (1) of the main paper) of a sphere with the

Layer	k	s	chns	input
$\mathrm{CBR2D}_0$	3	1	3/64	RGB image
AP_0	-	2	64/64	$\mathrm{CBR2D}_0$
$\mathrm{CBR2D}_1$	3	1	64/128	AP_0
$\mathrm{CBR2D}_2$	3	1	128/128	$\mathrm{CBR2D}_1$
AP_1	-	2	128/128	$\mathrm{CBR2D}_2$
$\mathrm{CBR2D}_3$	3	1	128/256	AP_1
$\mathrm{CBR2D}_4$	3	1	256/256	$\mathrm{CBR2D}_3$
AP_2	-	2	256/256	$\mathrm{CBR2D}_4$
$\mathrm{CBR2D}_5$	3	1	256/512	AP_2
$\mathrm{CBR2D}_6$	3	1	512/512	$\mathrm{CBR2D}_5$
$\mathrm{CBR2D}_7$	3	1	512/256	$\mathrm{CBR2D}_6$
UP_0	-	-	256/256	$\mathrm{CBR2D}_7$
$\mathrm{CBR2D}_8$	3	1	(256+256)/128	$\mathrm{CBR2D}_4$, UP_0
UP_1	-	-	128/128	$\mathrm{CBR2D}_8$
$\mathrm{CBR2D}_9$	3	1	(128+128)/64	$\mathrm{CBR2D}_2$, UP_1
UP_2	-	-	64/64	$\mathrm{CBR2D}_9$
Conv2D	3	1	(64+64)/16	$\mathrm{CBR2D}_0$, UP_2

Table 1: Architecture of "ImageFeaNet" inside the shape-from-shading network. "CBR2D" is a sequence consisting of 2D convolution ("Conv2D"), batch normalization, and ReLU activation. "AP" and "UP" are average pooling and bilinear upsampling, respectively, that double or halve the size of an input feature map. "k", "s", and "chns" denote the kernel size, the stride, and the numbers of input and output channels, respectively. The architecture of "AggregationNet" is the same as "ImageFeaNet" except the numbers of input and output channels. The number of the output channels of "AggregationNet" is 32.

Layer	k	s	chns	input
$\mathrm{CLR3D}_0$	1	1	64/64	Input Feature
$\mathrm{CLR3D}_1$	1	1	64/64	$\mathrm{CLR3D}_0$
$\mathrm{CLR3D}_2$	1	1	64/64	$\mathrm{CLR3D}_1$
$\mathrm{Conv3D}_3$	1	1	64/32	$\mathrm{CLR3D}_2$

(a) "3D Conv"

Layer	k	s	chns	input
$\mathrm{CBR3D}_0$	3	1	32/8	Input Feature
AP_0	-	2	8/8	$\mathrm{CBR3D}_0$
$\mathrm{CBR3D}_1$	3	1	8/16	AP_0
$\mathrm{CBR3D}_2$	3	1	16/16	$\mathrm{CBR3D}_1$
AP_1	-	2	16/16	$\mathrm{CBR3D}_2$
$\mathrm{CBR3D}_3$	3	1	16/32	AP_1
$\mathrm{CBR3D}_4$	3	1	32/32	$\mathrm{CBR3D}_3$
AP_2	-	2	32/32	$\mathrm{CBR3D}_4$
$\mathrm{CBR3D}_5$	3	1	32/64	AP_2
$\mathrm{CBR3D}_6$	3	1	64/64	$\mathrm{CBR3D}_5$
$\mathrm{CBR3D}_7$	3	1	64/32	$\mathrm{CBR3D}_6$
UP_0	-	-	32/32	$\mathrm{CBR3D}_7$
$\mathrm{CBR3D}_8$	3	1	32/16	$\mathrm{CBR3D}_4$ + UP_0
UP_1	-	-	16/16	$\mathrm{CBR3D}_8$
$\mathrm{CBR3D}_9$	3	1	16/8	$\mathrm{CBR3D}_2$+UP_1
UP_2	-	-	8/8	$\mathrm{CBR3D}_9$
Conv3D	3	1	8/1	$\mathrm{CBR3D}_0$ + UP_2

(b) "3D UNet"

Table 2: Architecture of "3D Conv" and "3D UNet" inside the cost volume filtering network. "CBR3D" is a 3D version of "CBR2D" and "CLR3D" is a sequence consisting of 3D convolution ("Conv3D") and leaky ReLU activation.

estimated BRDF under the same environment corresponds to one of the observed images. We repeat the alternating estimation 40 times.

We run the alternating estimation on a NVIDIA Tesla V100 (16GB). The update of geometry and reflectance takes approximately 5.5 minutes for each iteration. Thus the running time is approximately 4 hours in total. Note that most of the time is spent on the reflectance estimation and inference by the two deep neural networks (*i.e.*, the shape-from-shading network and the cost volume filtering network) takes milliseconds.

4.4. Data Augmentation of Measured BRDFs

As we explained in Sec. 4 of the main paper, we use measured BRDFs of the MERL BRDF database [11] to render synthetic training and test images. As the number of the BRDFs is small (94 for training), we augmented them by leveraging the conditional iBRDF model [4] as a generative model. We trained the conditional iBRDF model with the 94 measured BRDFs and sampled 2685 BRDFs from its latent space. We use the sampled BRDFs along with 2685 synthetic shapes [16] and 2685 captured illumination maps [6, 19] to synthesize the training set of the nLMVS-Synth dataset.

4.5. Evaluation

Evaluation of Recovered Mesh Models As described in Sec. 4.1 of the main paper, Tab. 2(b) of the main paper shows root-mean-square (RMS) of the distance from a point on the reconstructed mesh to the nearest point on the ground truth mesh. Since it is meaningless to compute errors for points that are invisible from the input views, we only used points on the recovered surface that are visible from at least three viewpoints to compute the RMS errors.

Comparison with Existing Methods For the qualitative and quantitative comparisons with the existing methods [17, 3, 18, 20, 21] reported in the main paper, we used the official implementations by the authors. As CVP-MVSNet [17], MVSNeRF [3], IDR [18], and NeRS [20] take standard dynamic range (SDR) images as inputs, we convert the high dynamic range (HDR) images of the nLMVS-Synth dataset into SDR images by applying gamma correction and clipping. We adjust a threshold of the clipping so that the total number of saturated and dark pixels is minimized.

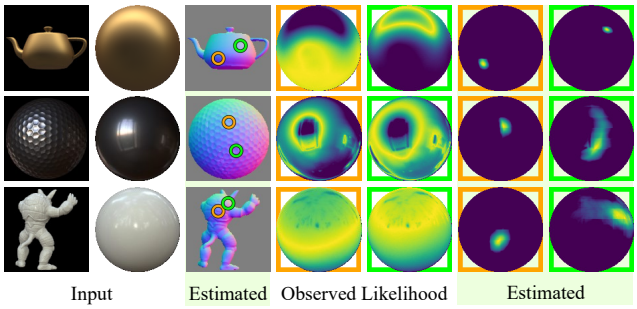


Figure 7: Results of the shape-from-shading network. The network successfully recovers a well-defined probability density of surface normals for each pixel and view.

3D Visualization In Fig. 1 of the main paper, for direct comparison with the results of IDR [18], we visualize the 3D reconstruction results of CVP-MVSNet [17], MVS-NeRF [3], and our method as (partial) 3D mesh models. Similar to the whole 3D shape recovery (Sec. 3.4 of the main paper), we reconstruct the mesh models from the estimated depth and surface normals by converting them into oriented points and applying Poisson surface reconstruction [8]. As CVP-MVSNet [17] and MVSNeRF [3] do not recover surface normals, we compute surface normal for each 3D point using its ten neighboring points.

5. Results on Synthetic Data

5.1. Accuracy of the Shape-from-Shading network

Figure 7 shows example estimation results of the shape-from-shading network. Surface normals are successfully recovered for different shapes and materials and the fundamental ambiguity of single-view recovery is represented well with the probability densities. Table 3a shows quantitative accuracy for each shape. The results are comparable to those of Johnson and Adelson [7] and the single-view method of Oxholm and Nishino [13]. Table 3b shows quantitative accuracy for each stage of the coarse-to-fine estimation. The results clearly show the effectiveness of the coarse-to-fine estimation. Table 3c shows how the grid size for the coarse-to-fine estimation affects the estimation accuracy and GPU memory size required for training. The results show that increasing the grid size improves the estimation accuracy, but also significantly increases the memory footprint.

In Tab. 3d, we show the effectiveness of modeling the observation likelihood as the Laplace distribution (Eq. (2) in the main text). We evaluate its effectiveness by comparing our method with its variant that models the likelihood as the Gaussian distribution. The results show the importance of modeling the observation likelihood as the Laplace distribution.

5.2. Joint Shape and Reflectance Estimation

Figures 8 to 10 show qualitative results of the joint estimation of geometry (depths and surface normals) and reflectance from images of five views. While existing methods struggle to recover accurate geometry for textureless, non-Lambertian objects, our method successfully recovers geometry and reflectance for different shapes, materials, and illumination environments.

Figure 11 shows two failure cases. As we can see in the first row, the"Golfball" object is relatively challenging for multi-view stereo as its geometry is repetitive, *i.e.*, difficult to find correspondences across views. As we can see in the second row, we also found that the BRDF estimation accuracy with the "Beige-Fabric" material is relatively poor. This is because the specular components of the estimated BRDF cannot be eliminated when it does not cause significant changes to the radiance of most of the surface points (*i.e.*, different surface normals). Evaluating such subtle difference between the rendered and the actual radiance without accurate geometry is a challenging problem.

5.2.1 Ablation Study

Effectiveness of The Probabilistic Representation of Surface Normals We show the effectiveness of the probabilistic representation of surface normals by comparing our method with that without the probability density function ("w/o PDF") which only uses the surface normals with the highest probability (*i.e.*, "ArgMax Normal" in Fig. 6) as inputs to the cost volume filtering network. Ground-truth BRDFs were used to ablate just the probabilistic representation. Table 4 shows quantitative results. The results show that the probabilistic representation improves the accuracy, especially for the most complex shape, Armadillo ("A"). As we have discussed in Sec. 3.1 of the main paper, the probabilistic representation is vital to resolve the ambiguity arising in shape-from-shading under natural illumination.

Effectiveness of The Loss functions for Training We show the effectiveness of the depth supervision (Eq. (8)), the normal supervision (Eq. (9)), and the consistency loss (Eq. (7)) by ablating them during the training of the cost volume filtering network. Similar to the ablation study on the probabilistic representation of surface normals, we used ground-truth BRDFs as inputs to the network. Table 5 shows quantitative results. The results clearly show the effectiveness of each loss function.

Number of Input Views As the cost volume filtering network inside our nLMVS-Net can take images of an arbitrary number of input views, we evaluate how the number of input views affects the estimation accuracy. For 6 combinations of shapes, materials, and illuminations, we varied

	Mat.	Shape	$\leq$ 10 deg	Median
Ours		S	99.7%	3.1 deg
		T	91.3%	3.7 deg
		G	84.7%	7.3 deg
		P	85.8%	7.2 deg
		B	75.6%	12.4 deg
		A	40.7%	24.2 deg
JA [7]	L	blobs	90%	N/A
ON [13]	nL	blobs	N/A	15 deg

(a) Accuracy for each shape

	$\leq$ 10 deg	Median	Mean
Likelihood	56.6%	49.4 deg	19.8 deg
1st	75.9%	26.9 deg	9.4 deg
2nd	80.3%	26.5 deg	8.4 deg
Final (Ours)	83.4%	24.2 deg	7.7 deg

(b) Accuracy for each stage

Grid Size	Mem.	$\leq$ 10 deg	Median	Mean
2×2	6.6 GB	74.2%	31.5 deg	10.0 deg
4×4	16.6 GB	79.7%	41.3 deg	8.9 deg
8×8 (ours)	43.5 GB	83.4%	24.2 deg	7.7 deg

(c) Effectiveness of the grid size

Likelihood Distribution	$\leq$ 10 deg	Median	Mean
Gaussian	77.7%	30.2 deg	9.0 deg
Laplace (Ours)	83.4%	24.2 deg	7.7 deg

(d) Effectiveness of using the Laplace distribution

Table 3: Accuracy of the shape-from-shading network. "$\leq$ 10 deg" is the ratio of pixels whose surface normal estimation error is lower than 10 degrees. "Median" and "Mean" are median and mean errors of the estimated surface normals. Note that, for direct comparison with the single-view method of Oxholm and Nishino [13], we computed the median error for each image and report the worst case. (a) Accuracy for each shape. We also show the results of Johnson and Adelson (JA) [7] and the single-view method by Oxholm and Nishino (ON) [13]. The second and the third columns denote the reflectance ("nL" for non-Lambertian and "L" for Lambertian) and shape used in the evaluations. The results are comparable to these existing methods. (b) Accuracy for each stage of the coarse-to-fine estimation. "Likelihood" is the accuracy of surface normals with the highest observation likelihood. The results clearly show the effectiveness of the proposed coarse-to-fine architecture. (c) We show how the grid size for the coarse-to-fine estimation affects estimation accuracy and GPU memory size ("Mem.") required for training. Increasing the grid size improves the estimation accuracy, but also significantly increases the memory footprint. Note that we couldn't increase the grid size beyond 8×8 due to limitation of GPU memory. (d) Effectiveness of modeling the observation likelihood as the Laplace distribution (Eq. (2) in the main text). We evaluate its effectiveness by comparing our method with its variant that models the likelihood as the Gaussian distribution. The results show the importance of modeling the observation likelihood as the Laplace distribution.

the number of input views from 3 to 9 and computed estimation errors of depths, surface normals, and reflectance. Table 6 shows quantitative results. While our method can recover geometry even from 3 view inputs, using 5 view input images improves the estimation accuracy. Although it is possible to use images of more than five views as inputs, it didn't improve the estimation accuracy. This could be due to the gap between the numbers of input views used for training (3) and inference (7 or 9).

5.3. Whole 3D Shape Recovery

Figure 12 shows 3D mesh models recovered from 10 view images of the nLMVS-Synth dataset. As we recover the mesh models by simply applying Poisson surface reconstruction (PSR) [8], silhouettes of our results are sometimes inconsistent with the input views. For this reason, our method with PSR [8] does not quantitatively outperform IDR [18] that explicitly imposes consistency of silhouettes. Our results for non-Lambertian objects, however, are qualitatively more plausible than those of other existing methods. Please see, for instance, the legs of the Stanford bunny in Fig. 12. This shows the advantage of our method in recovering surface details for textureless, non-Lambertian surfaces.

In Fig. 12, we also compare our results with those of NeuS [15], a state-of-the-art neural image synthesis method. Although this method can work without input silhouette images, it often fails to recover surface geometry from the sparse (*i.e.*, 10 view) inputs as the optimization of geometry and appearance without silhouette images is unstable.

6. Results on The Multiview Objects Under Natural Illumination Database [12]

Figure 13 shows whole 3D shape and appearance recovered from our estimation results on the Multiview Objects Under Natural Illumination Database [12]. We compare our results with those of Oxholm and Nishino [13]. Our results

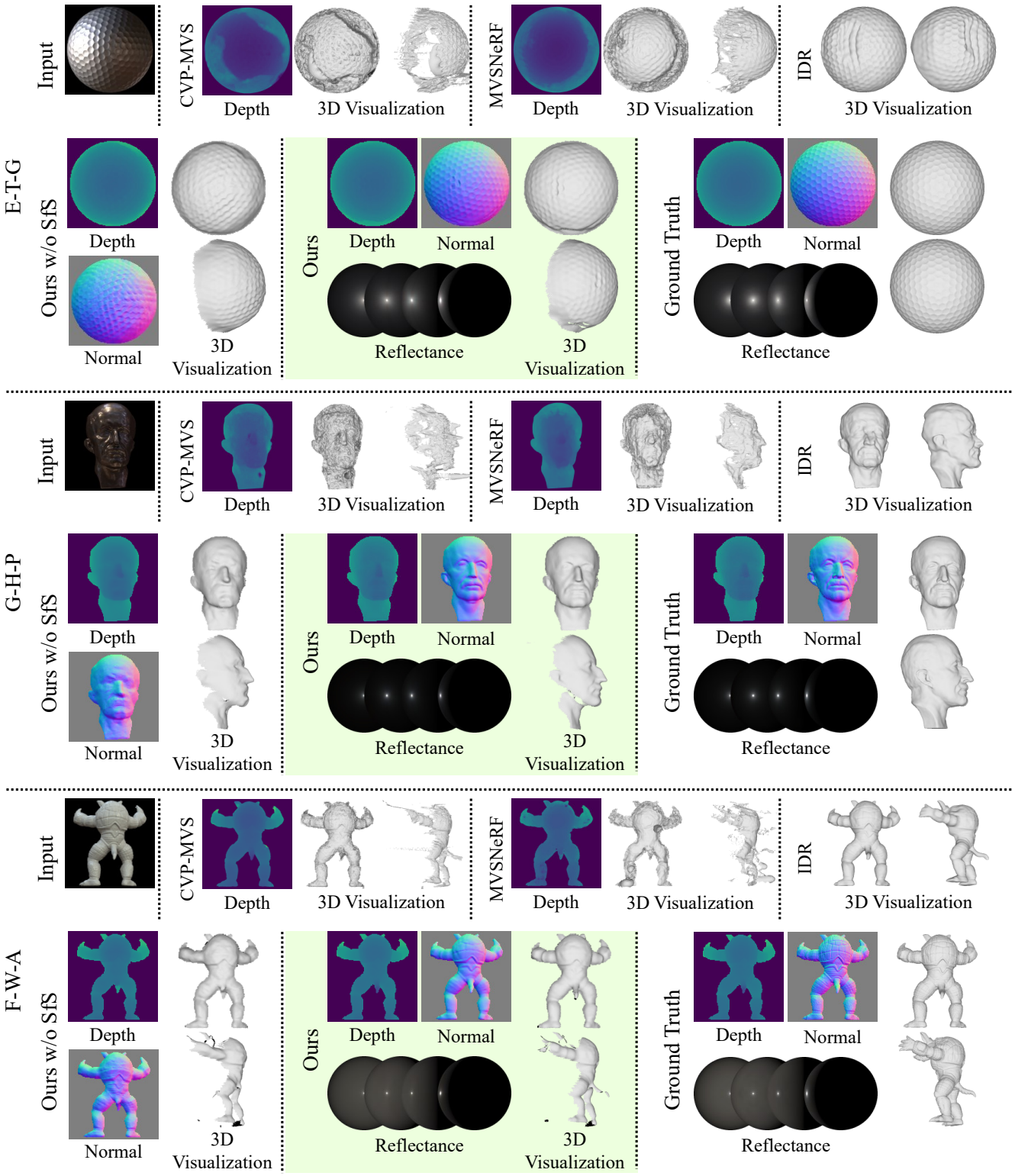


Figure 8: Estimated geometry (depths and surface normals) and reflectance from five view images of the nLMVS-Synth dataset. The initials on the left side of the images correspond to names of the environment, the material, and the shape. While existing methods struggle to handle textureless, non-Lambertian objects, our method can successfully recover surface normals and depth for objects with different shapes and materials.

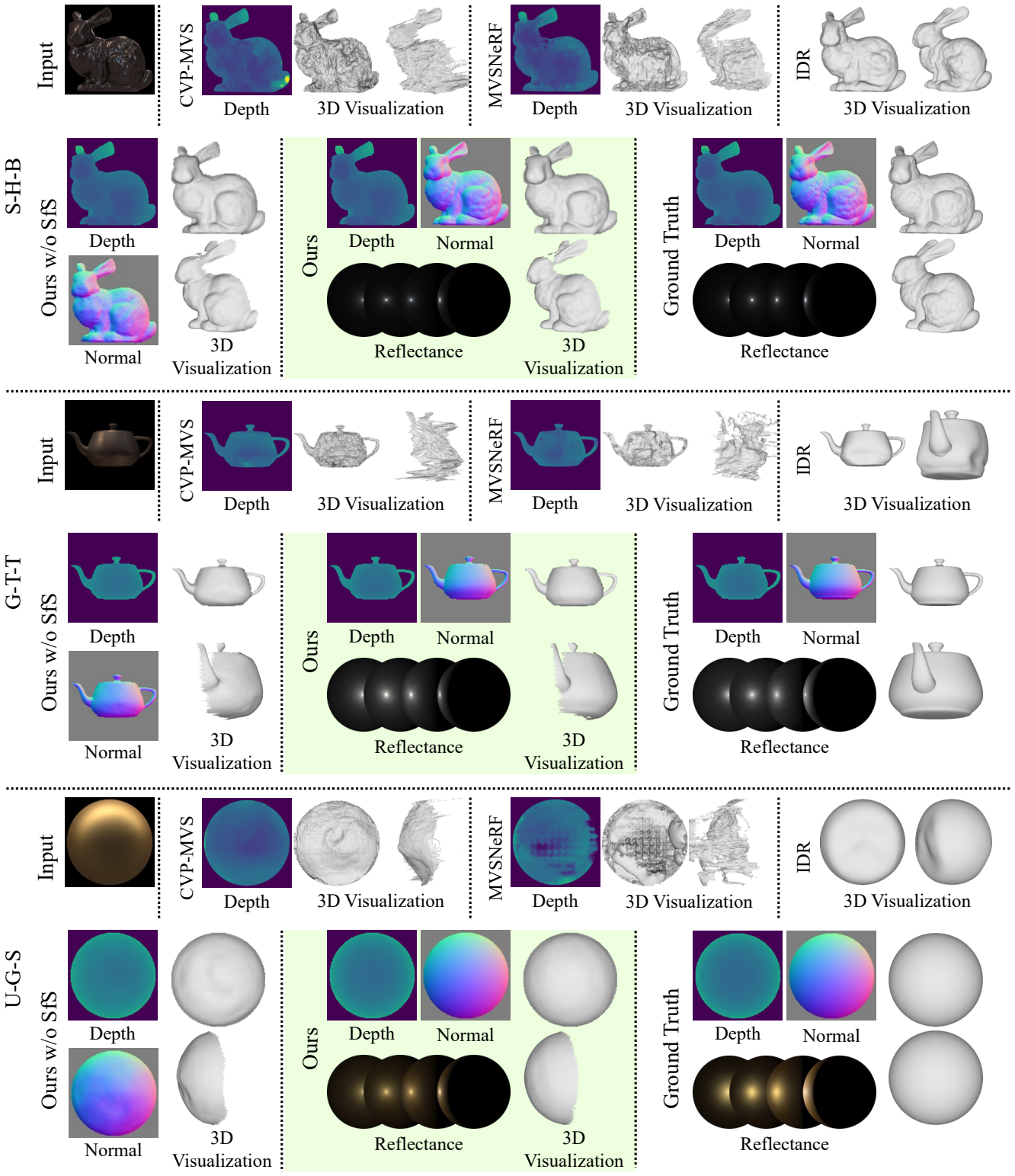


Figure 9: Estimated geometry (depths and surface normals) and reflectance from five view images of the nLMVS-Synth dataset. The initials on the left side of the images correspond to names of the environment, the material, and the shape. While existing methods struggle to handle textureless, non-Lambertian objects, our method can successfully recover surface normals and depth for objects with different shapes and materials.

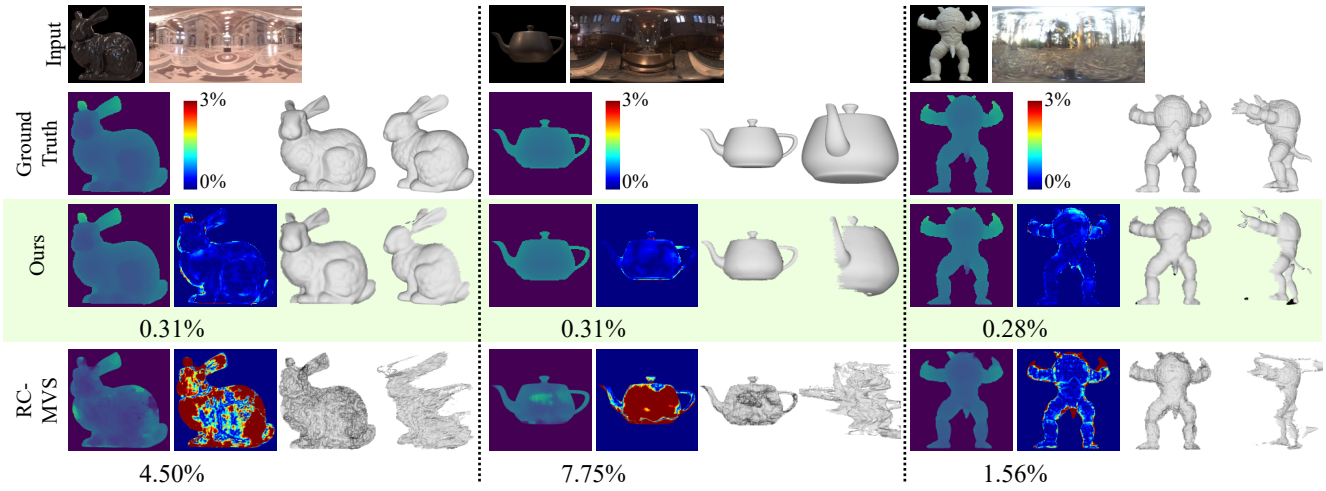


Figure 10: Qualitative comparison with Chang *et al*. [2] (RC-MVSNet). While RC-MVSNet does not require any ground-truth 3D information during training (*i.e*., it is an unsupervised method), it fundamentally relies on a photometric consistency loss based on the Lambertian assumption. For this, although they improved the robustness to non-Lambertian surfaces by introducing a view synthesis loss, their method fails on textureless objects whose appearance is completely different from Lambertian.

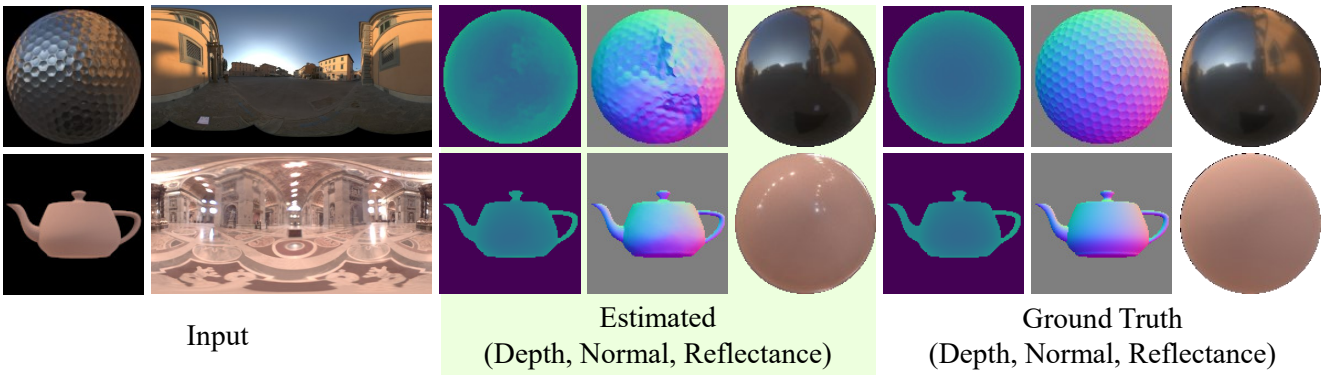


Figure 11: Failure cases. In this figure, we visualize the estimated and the ground truth BRDFs as images of spheres rendered under the same illumination environment. The "Golfball" (the first row) object is challenging for multi-view stereo as its geometry is repetitive, *i.e*., difficult to find correspondences across views. We can also see that the BRDF estimation accuracy for the objects with the "Beige-Fabric" material (the second row) is also relatively poor. This is because the specular components of the estimated BRDF cannot be eliminated when it does not cause significant changes to the radiance of most of the surface points (*i.e*., different surface normals). Evaluating such subtle difference between the rendered and the actual radiance without accurate geometry is a challenging problem.

have fewer artifacts and more surface details (*e.g*., ears of the "Horse") are recovered. Note that, although ground-truth 3D mesh models are provided by the authors of the dataset [12], quantitative evaluation that accurately reflects superiority of each method is difficult due to the poor geometric calibration of the dataset.

Figure 14 shows estimated BRDFs. The results are consistent across two different environments. Note that we couldn't run our method on images of the mirror "Pig" object under the "main" environment and those captured under the "Outdoor" environment as they are strongly affected by the flaws of the dataset, namely the saturation of specular highlights. Note also that the "Pig" object was painted after the image capture under the "Hall" environment and looks different in the images captured under the "Indoor" environment.

7. Results on the nLMVS-Real Dataset

Comparison with IDR [18] We compare our method with IDR [18] on several of the real-world images of

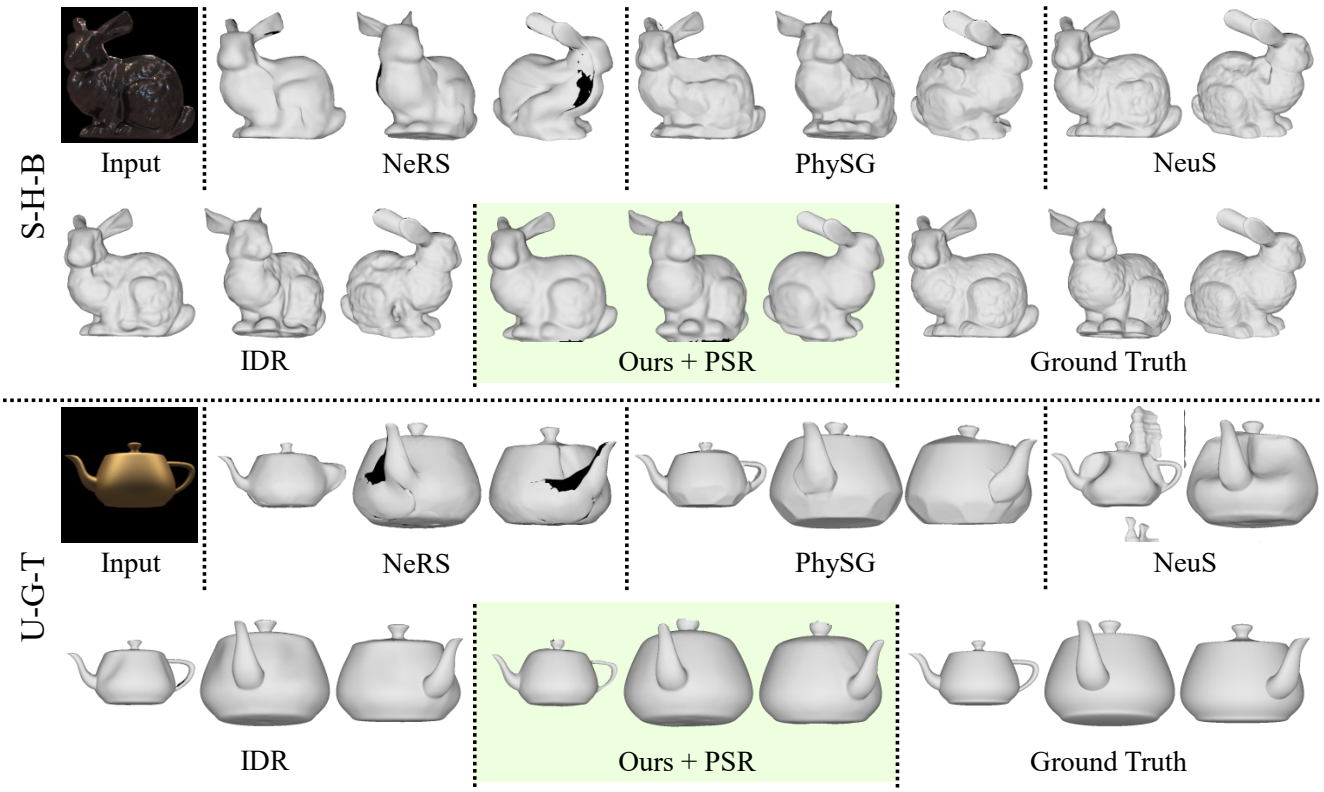


Figure 12: Recovered mesh models from 10 view images of the nLMVS-Synth dataset. Although our method with Poisson surface reconstruction (PSR) [8] does not quantitatively outperform IDR [18], our results for objects with complex, non-Lambertian materials have fewer artifacts. This shows the advantage of our method in recovering surface details for textureless, non-Lambertian objects.

	Depth Error		Normal Error	
	Ours	w/o PDF	Ours	w/o PDF
S	0.29%	0.29%	1.9 deg	1.8 deg
T	0.38%	0.38%	4.3 deg	4.3 deg
G	0.21%	0.20%	5.1 deg	5.2 deg
P	0.25%	0.26%	4.9 deg	5.0 deg
B	0.36%	0.38%	6.7 deg	6.9 deg
A	0.55%	0.61%	14.6 deg	15.4 deg
Ave.	0.34%	0.35%	6.3 deg	6.4 deg

Table 4: Ablation study on the probabilistic surface normal representation. Ours "w/o PDF" (without probability density function) only uses the surface normals with highest probability as inputs to the cost volume filtering network. Each row shows reconstruction errors for each shape. The results show that the probabilistic representation improves the accuracy, especially for the most complex shape, Armadillo ("A"). Note that ground-truth BRDFs were used to ablate just the probabilistic representation.

	Depth	Normal
w/o Depth Supervision	27.74 %	7.1 deg
w/o Normal Supervision	0.49 %	9.0 deg
w/o Consistency Loss	0.35 %	6.3 deg
Ours	0.34 %	6.3 deg

Table 5: Ablation study on the loss functions for the training of the cost volume filtering network. The results clearly show the effectiveness of each loss function.

our nLMVS-Real dataset. Similar to comparisons on the nLMVS-Synth dataset, we used images of five views (a reference view and four neighboring views) as inputs for this evaluation. Figure 15 shows qualitative and quantitative results. As we assume that the illumination is distant and shadows and interreflections are negligible, our results are relatively poor in regions where these assumptions do not hold, *e.g.*, the lower jaw and the bottom of the "Horse" object. Our results for the top half of the objects are, however, quantitatively and qualitatively more accurate than IDR [18]. This demonstrates the effectiveness of our method. Note also that, different from IDR [18], we esti-

	# of Views			
	3	5	7	9
E-T-G	0.54	0.47	0.46	0.48
G-H-P	0.44	0.37	0.33	0.32
F-W-A	0.31	0.28	0.25	0.26
S-H-B	0.37	0.31	0.32	0.31
G-T-T	0.41	0.31	0.40	0.37
U-G-S	0.95	0.59	0.67	0.93
Avg.	0.50	0.39	0.41	0.44

(a) Depth Errors (%)

	# of Views			
	3	5	7	9
E-T-G	6.7	6.7	6.8	6.8
G-H-P	6.2	5.9	5.7	5.7
F-W-A	10.4	10.2	10.1	10.3
S-H-B	7.3	6.9	7.3	7.5
G-T-T	4.9	4.2	4.1	4.2
U-G-S	2.7	2.3	2.3	2.4
Avg.	6.4	6.0	6.1	6.2

(b) Normal Errors (degree)

Table 6: Relationship between the number of input views and estimation errors. The initials in the first column correspond to names of the environment, the material, and the shape. While our method can recover geometry even from 3 view inputs, using 5 view input images improves the estimation accuracy. Although it is possible to use images of more than five views as inputs, it didn't improve the estimation accuracy. This could be due to the gap between the numbers of input views used for training (3) and inference (7 or 9).

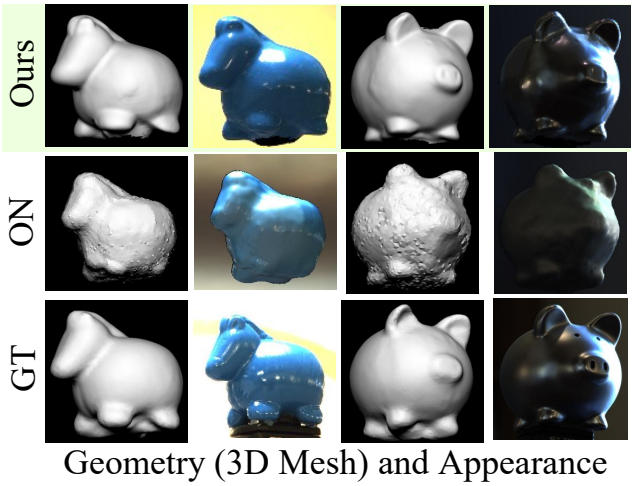


Figure 13: 3D mesh models and their appearance recovered from our estimation results on the Multiview Objects Under Natural Illumination Database [12]. Compared to the results of Oxholm and Nishino (ON) [13], our results have fewer artifacts and surface details are successfully recovered.

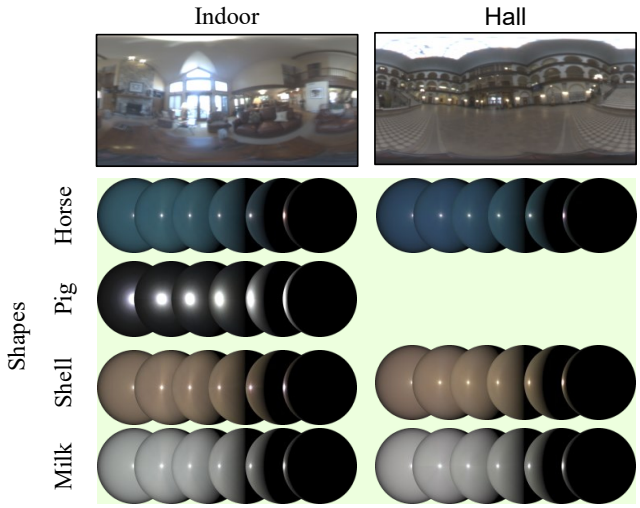


Figure 14: BRDF estimation results on the Multiview Objects Under Natural Illumination Database [12]. The results are consistent across two different environments.

mate not only surface geometry but also reflectance of the objects, which can be used for applications such as relighting.

Qualitative Results and Discussions Figure 16 and Fig. 17 show estimated BRDFs. Figure 18, Fig. 19, Fig. 20, Fig. 21, Fig. 22, and Fig. 23 show recovered geometry from the nLMVS-Real dataset. For this experiment, we used all views (roughly 20 views) of the nLMVS-Real dataset as inputs to recover depths surface normals, reflectance, and mesh models. The results demonstrate the robustness of our method. As we reported in the main text, our method fails on approximately 30% of all input images. There are two factors that cause these failure cases. The first is that the assumption of distant and static illumination is sometimes not the case in real-world environments. If the intensity of sunlight drastically changes during the image capture due to movement of clouds, our method fails to recover accurate geometry and reflectance. Note, however, that our method successfully works with many real-world images captured under outdoor scenes. This shows the robustness of our method to small changes in the illumination environment. The second is that several combinations of materials and illumination environments are challenging. As we can see in Fig. 21, our method fails on the combination of the "Lab-

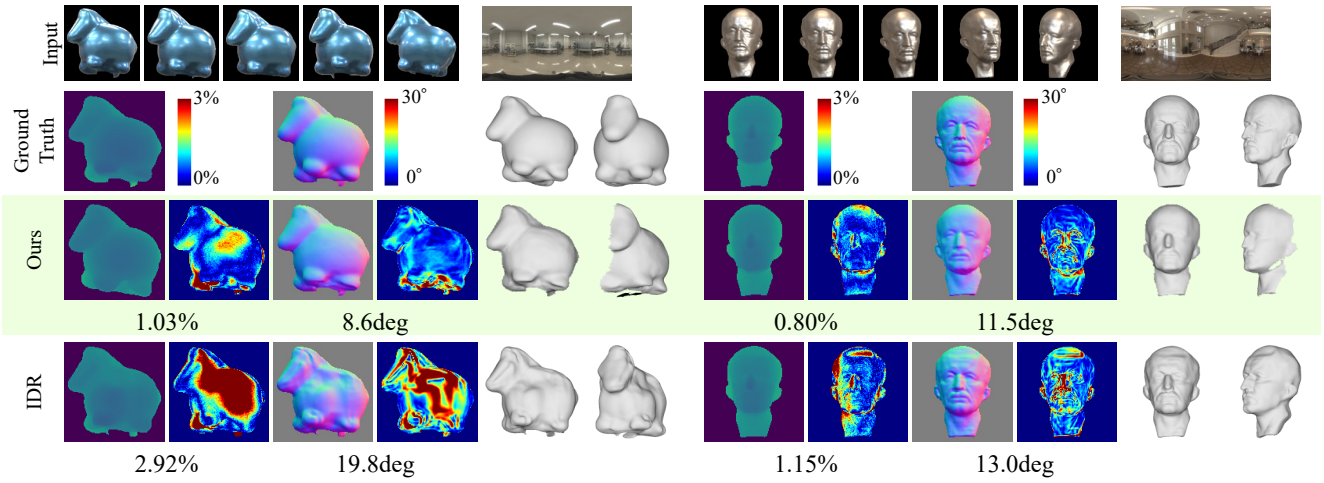


Figure 15: Comparison with IDR [18] on real-world images of nLMVS-Real dataset. The numbers are mean estimation errors. Please see the text for discussions.

oratory" environment and the "Duralumin" or the "White-Primer" material. These combinations are challenging as specular highlights from each light cannot be distinguished and diffuse shading does not exist (for the "Duralumin" material) or has the same color as the specular reflection (for the "White-Primer" material).

References

[1] John Burkardt. "PLY Files - an ASCII Polygon Format". https://people.sc.fsu.edu/~jburkardt/data/ply/ply.html.

[2] Di Chang, Aljaž Božič, Tong Zhang, Qingsong Yan, Yingcong Chen, Sabine Süsstrunk, and Matthias Nießner. RC-MVSNet: Unsupervised Multi-View Stereo with Neural Rendering. In *Proc. ECCV*, 2022.

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo. In *Proc. ICCV*, 2021.

[4] Zhe Chen, Shohei Nobuhara, and Ko Nishino. Invertible Neural BRDF for Object Inverse Rendering. In *Proc. ECCV*, 2020.

[5] Paul Debevec. "Light probe Image Gallery". https://www.pauldebevec.com/Probes/.

[6] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to Predict Indoor Illumination from a Single Image. *ACM TOG*, 36(6), 2017.

[7] Micah K. Johnson and Edward H. Adelson. Shape Estimation in Natural Illumination. In *Proc. CVPR*, pages 2553–2560, 2011.

[8] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[9] ICT Vision & Graphics Lab. "High-Resolution Light Probe Image Gallery". https://vgl.ict.usc.edu/Data/HighResProbes/.

[10] The Stanford Computer Graphics Laboratory. "The Stanford 3D Scanning Repository". http://www.graphics.stanford.edu/data/3Dscanrep/.

[11] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A Data-Driven Reflectance Model. *ACM TOG*, 22(3):759–769, 2003.

[12] Geoffrey Oxholm and Ko Nishino. Multiview Shape and Reflectance from Natural Illumination. In *Proc. CVPR*, pages 2163–2170, 2014.

[13] Geoffrey Oxholm and Ko Nishino. Shape and Reflectance Estimation in the Wild. *TPAMI*, 38(2):376–389, 2015.

[14] Szymon Rusinkiewicz, Doug DeCarlo, Adam Finkelstein, and Anothony Santella. "Suggestive Contour Gallery". https://gfx.cs.princeton.edu/proj/sugcon/models/.

[15] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *Proc. NeurIPS*, 2021.

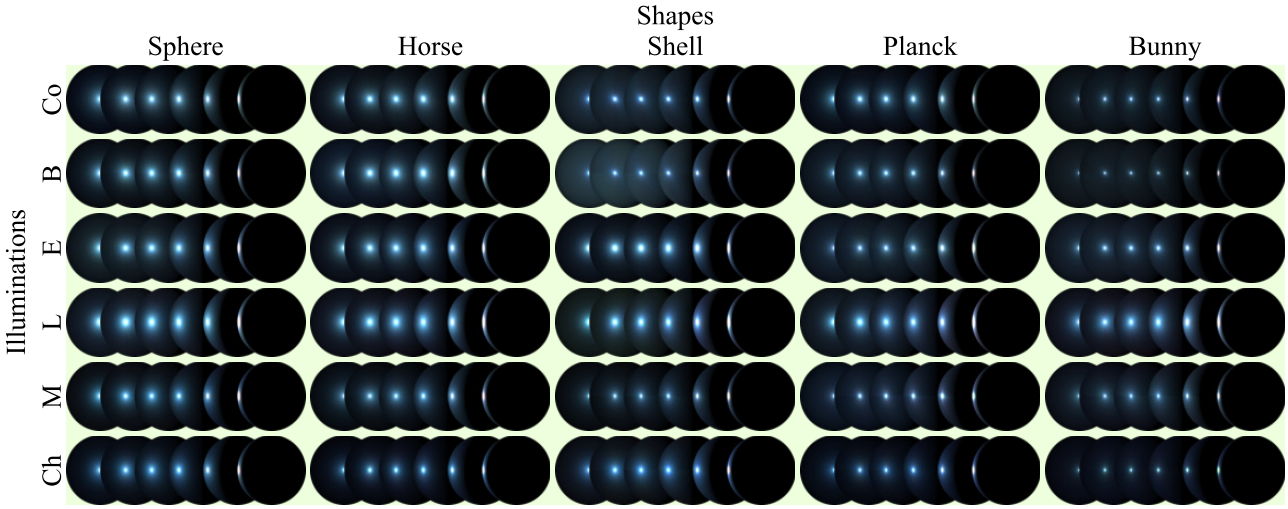
[16] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep Image-Based Relighting from Optimal Sparse Samples. In *Proc. SIGGRAPH*, 2018.

[17] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost Volume Pyramid Based Depth Inference for Multi-View Stereo. In *Proc. CVPR*, 2020.

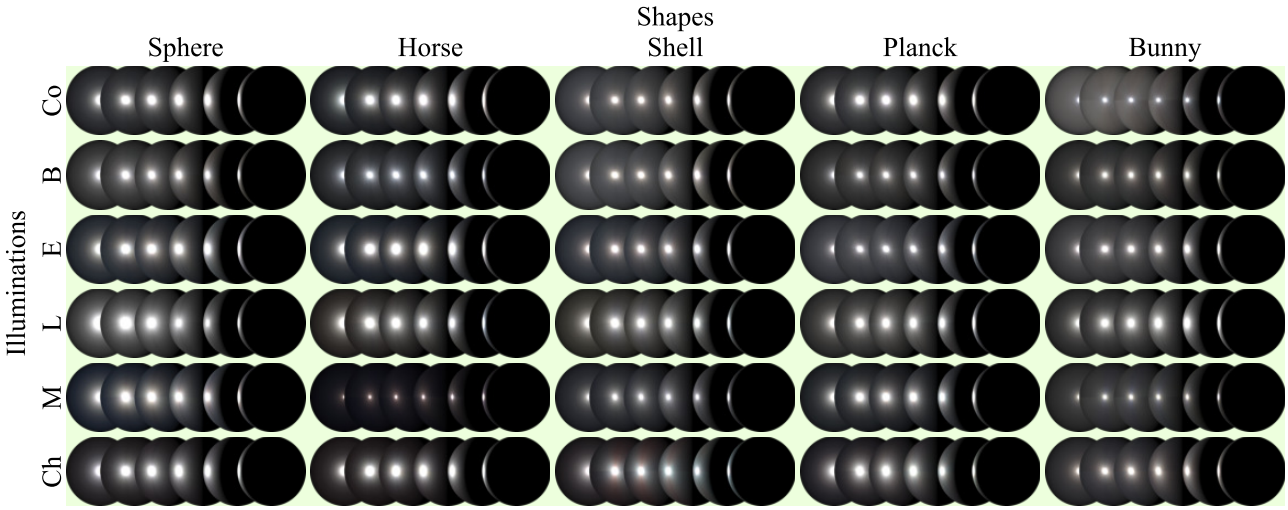
[18] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *Proc. NeurIPS*, 2020.

[19] Greg Zaal, Rob Tuytel, Rico Cilliers, James Ray Cock, Andreas Mischok, and Sergej Majboroda. "Poly Haven". https://polyhaven.com/hdris.

[20] Jason Y. Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural Reflectance Surfaces for



(a) Results for "Blue-Metallic" material.

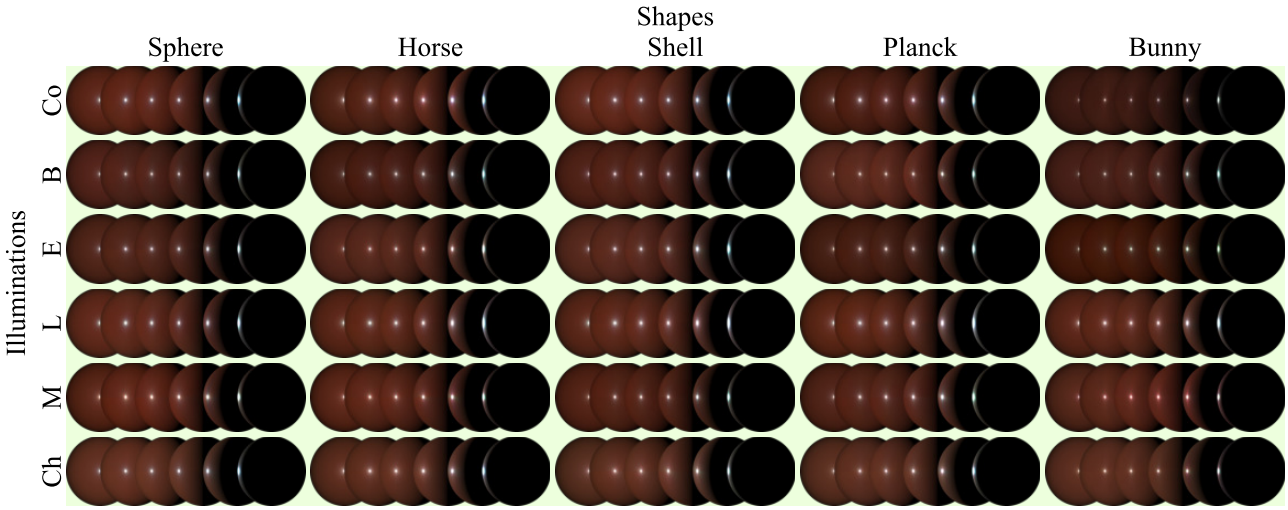


(b) Results for "Duralumin" material.

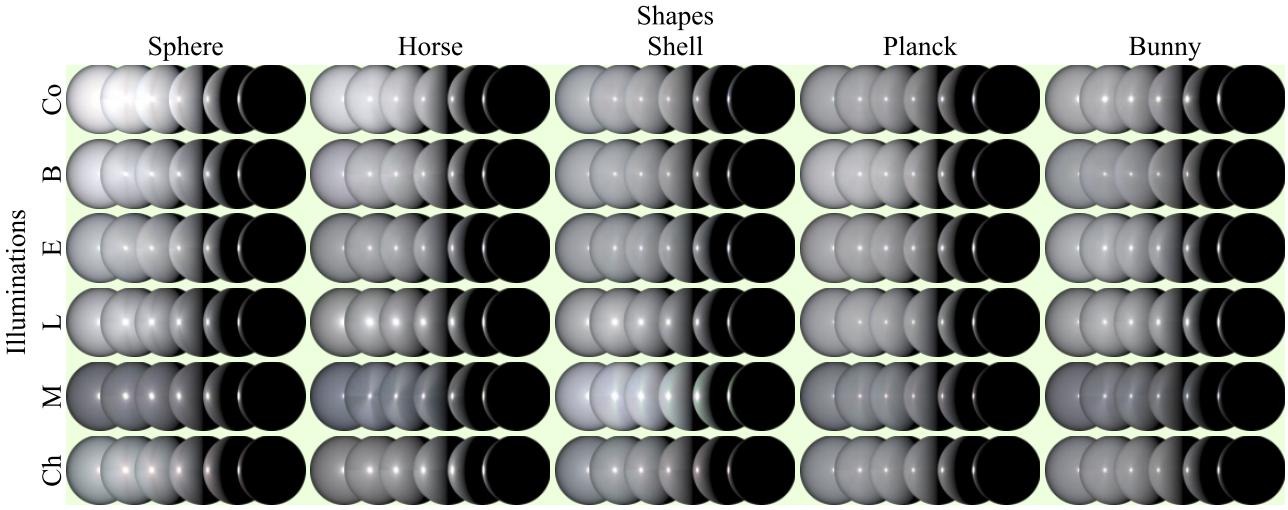
Figure 16: Estimated BRDFs from the nLMVS-Real dataset.

Sparse-view 3D Reconstruction in the Wild. In *Proc. NeurIPS*, 2021.

[21] 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 *Proc. CVPR*, 2021.



(a) Results for “Bright-Red” material.



(b) Results for “White-Primer” material.

Figure 17: Estimated BRDFs from the nLMVS-Real dataset.

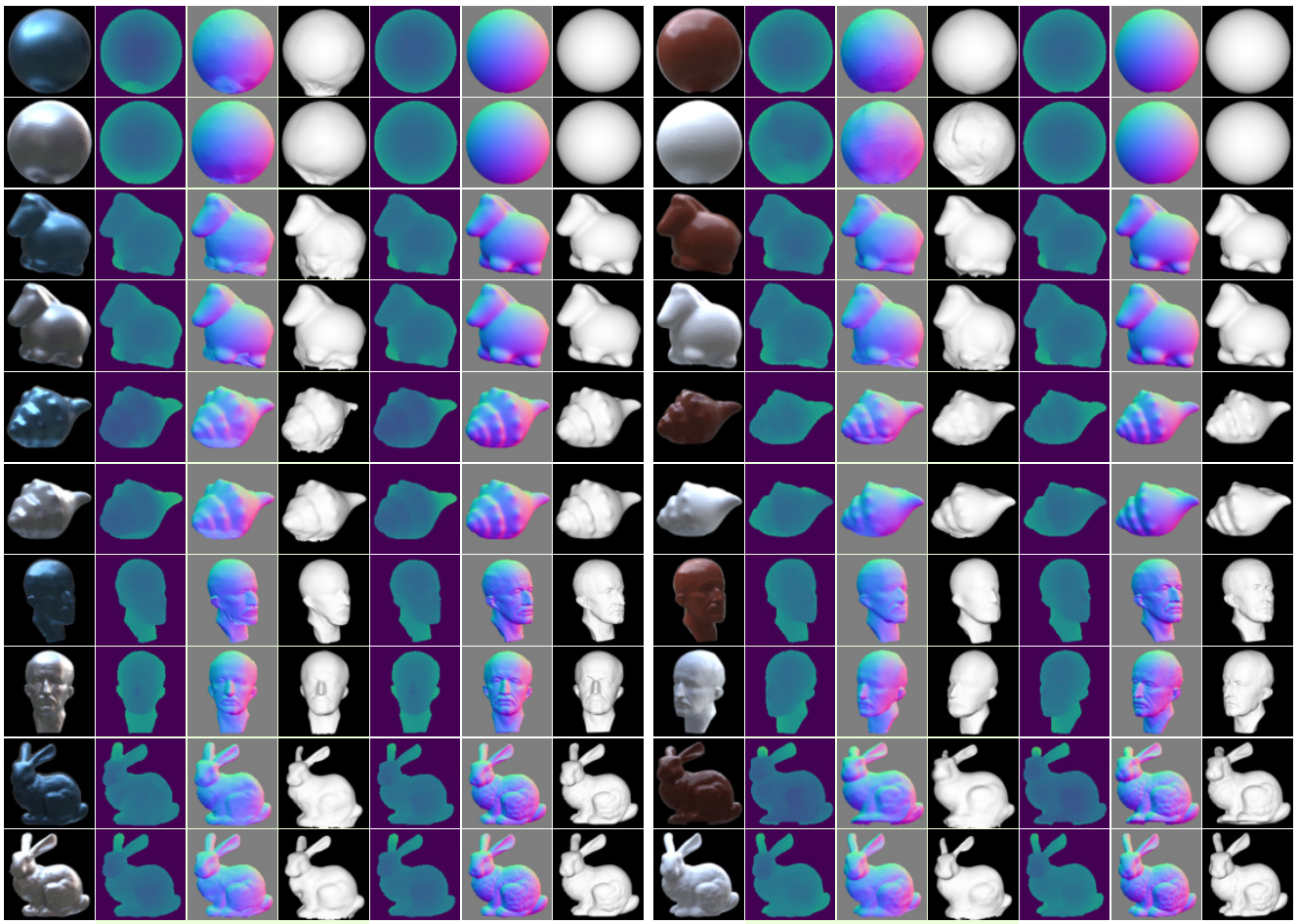


Figure 18: Estimated geometry from the nLMVS-Real dataset (Results under “Court” environment). Please see the text for discussions about failure cases.

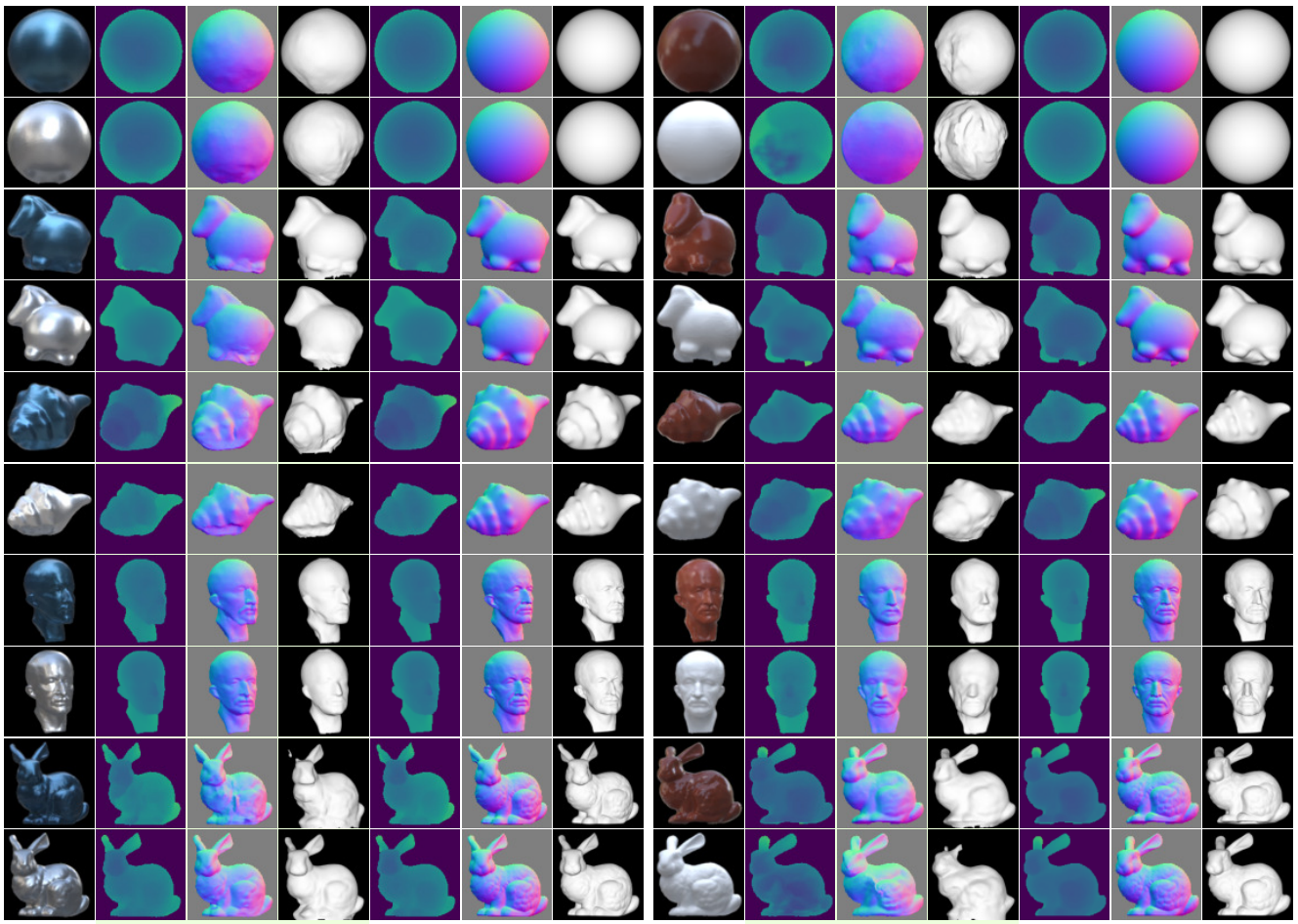


Figure 19: Estimated geometry from the nLMVS-Real dataset (Results under "Buildings" environment). Please see the text for discussions about failure cases.

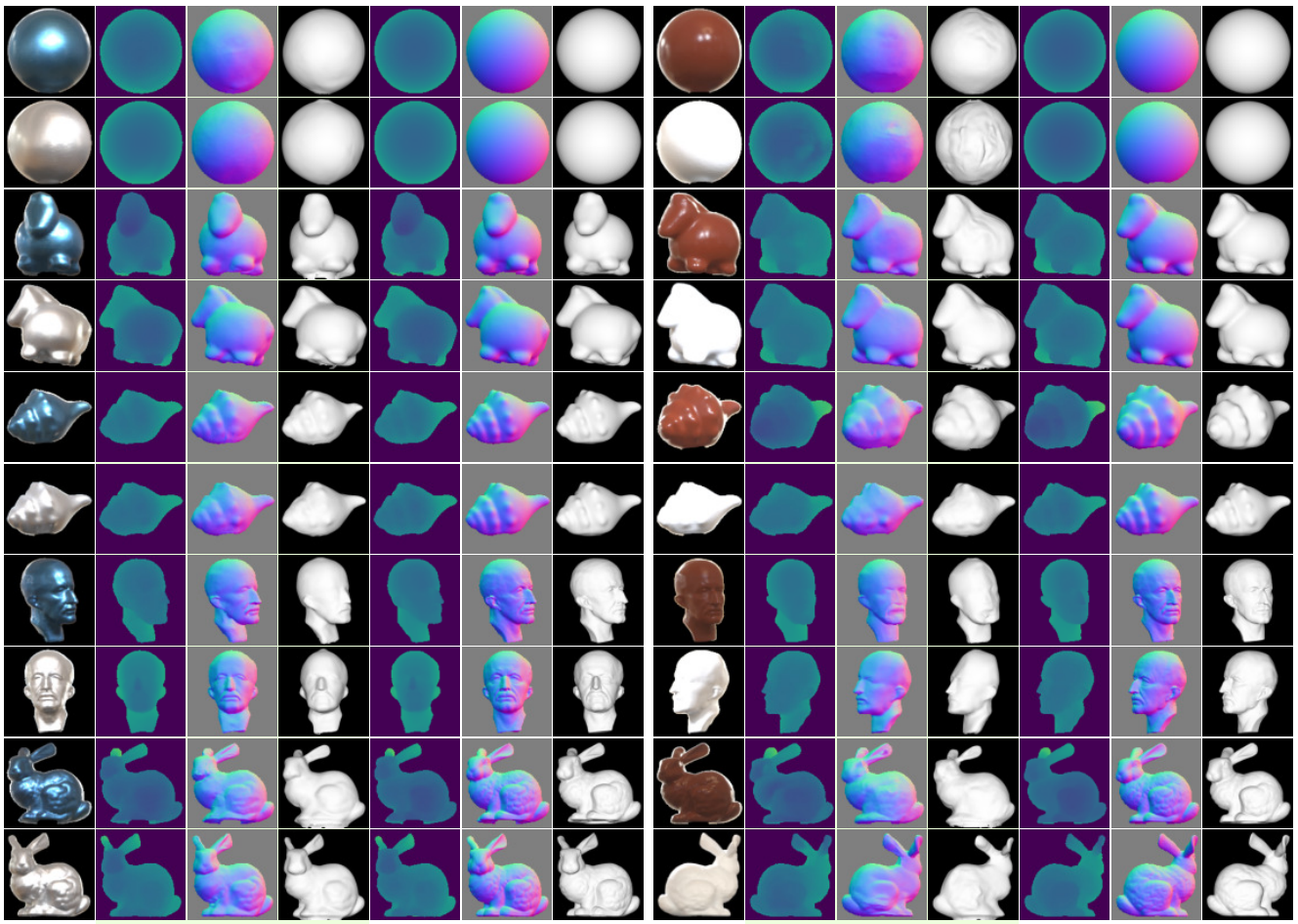


Figure 20: Estimated geometry from the nLMVS-Real dataset (Results under "Entrance" environment). Please see the text for discussions about failure cases.

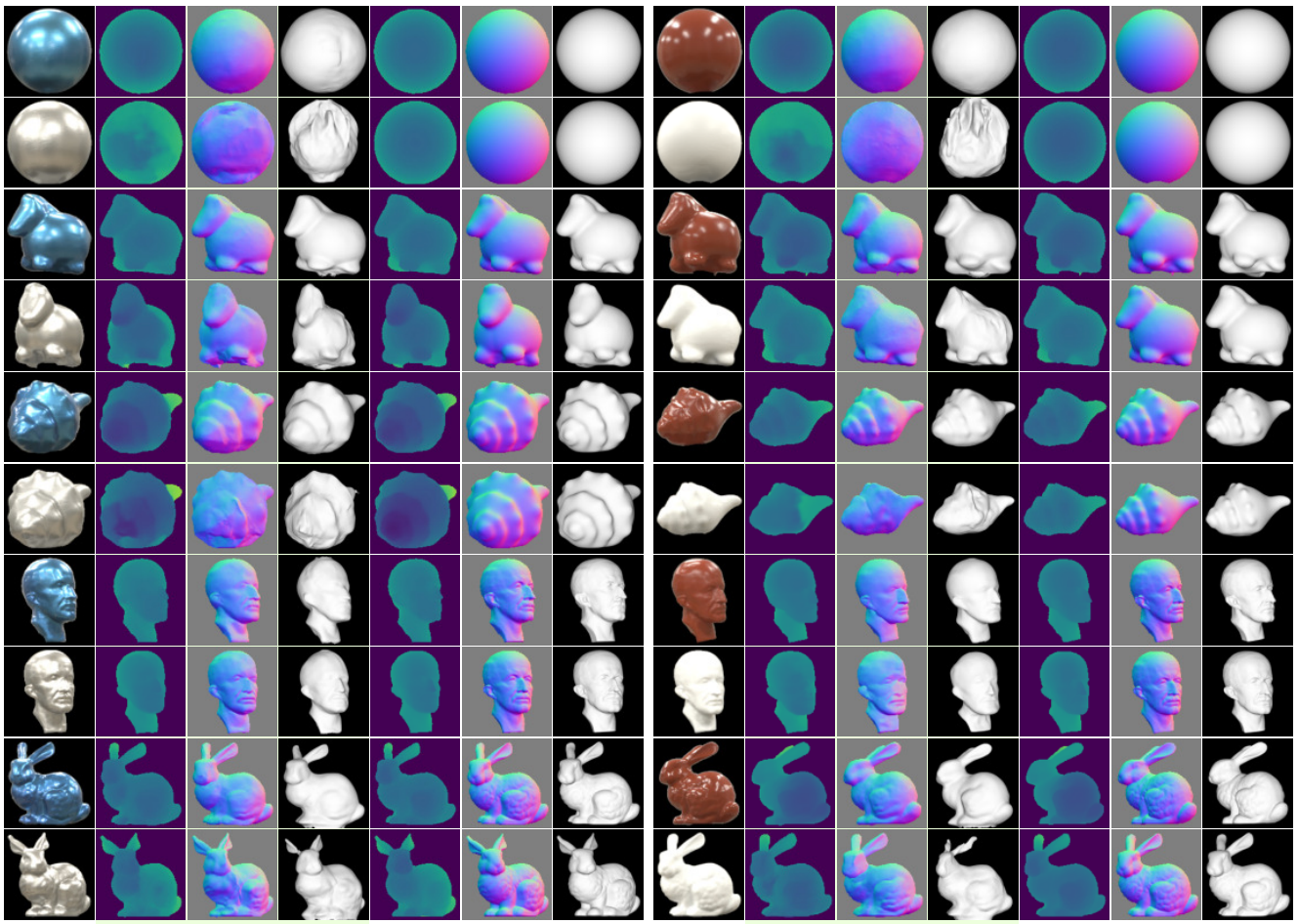


Figure 21: Estimated geometry from the nLMVS-Real dataset (Results under “Laboratory” environment). Please see the text for discussions about failure cases.

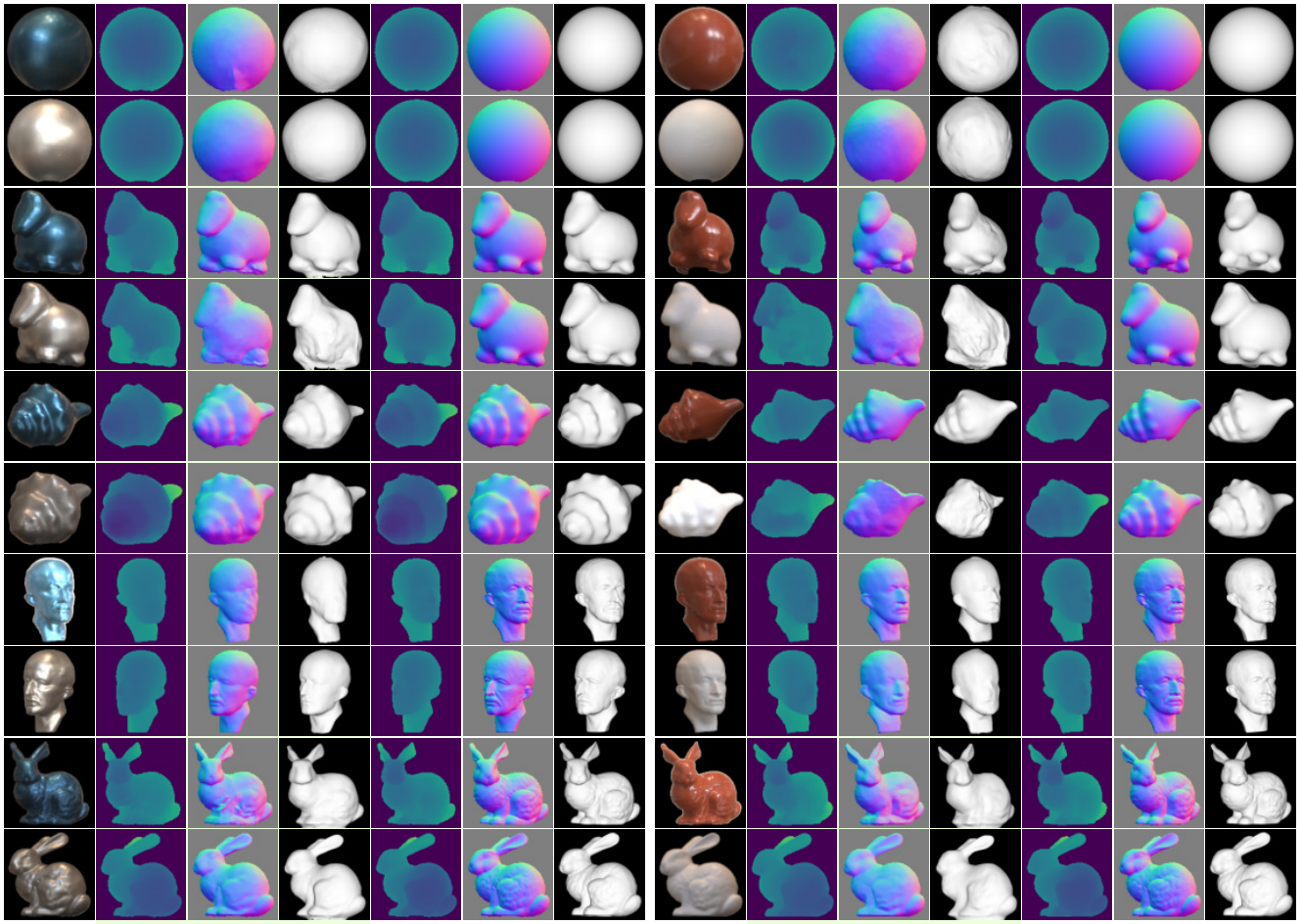


Figure 22: Estimated geometry from the nLMVS-Real dataset (Results under “Manor House” environment). Please see the text for discussions about failure cases.

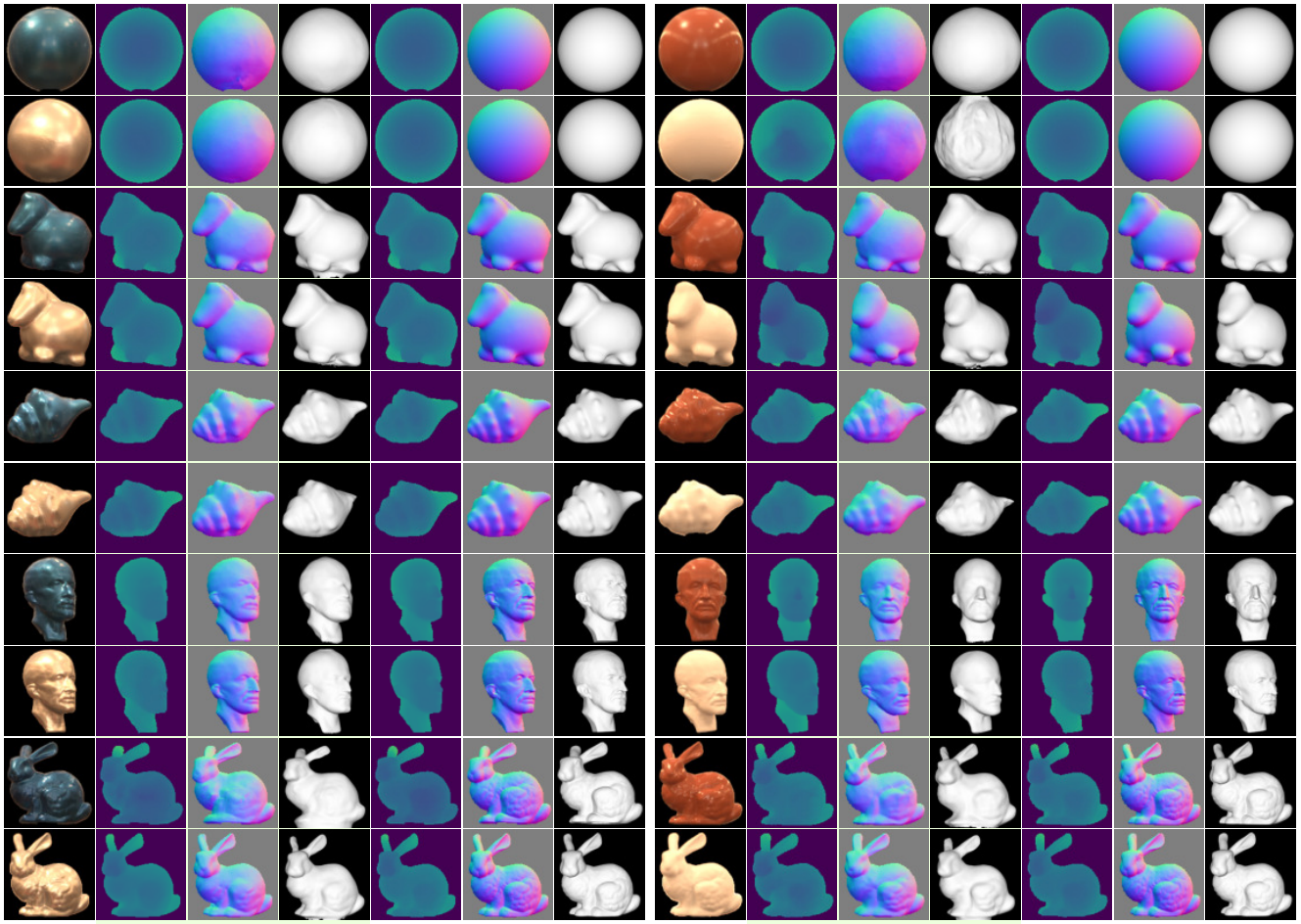


Figure 23: Estimated geometry from the nLMVS-Real dataset (Results under “Chapel” environment). Please see the text for discussions about failure cases.