NEURAL PHOTOMETRIC STEREO FOR SHAPE AND MATERIAL ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper addresses a challenging Photometric-Stereo problem where the object to be reconstructed has unknown, non-Lambertian, and possibly spatially-varying surface materials. This problem becomes even more challenging when the shape of the object is highly complex so that shadows cast on the surface are inevitable. To overcome these challenges, we propose a simple coordinate-based deep MLP (multilayer perceptron) neural network to parameterize both the unknown 3D shape and the unknown spatially-varying reflectance at every image pixel. This network is able to leverage the observed specularities and shadows on the surface, and recover both surface shape, normal and generic non-Lambertian reflectance via an inverse differentiable rendering process. We explicitly predict cast shadows, mitigating possible artifacts on these shadowing regions, leading to higher estimation accuracy. Our framework is entirely self-supervised, in the sense that it requires neither ground truth shape nor known svBRDF. Tests on real-world images demonstrate that our method achieves state-of-the-art accuracy in both shape recovery and material estimation. Thanks to the small size of the MLP-net, our method is also an order of magnitude faster than previous competing deep-learning based photometric stereo methods.

1 INTRODUCTION

Photometric Stereo is a classic technique aiming to recover the surface shape of an object from its multiple images taken at the same viewpoint but under different illuminations. It offers an effective way of reconstructing 3D shapes. When the surface reflectance is Lambertian (diffuse), the problem of photometric stereo can be solved in closed-form (Woodham, 1980). However, it remains a challenging task for non-Lambertian object recovery by Photometric stereo, where the appearance of the object contains certain view-dependent specularities, and is sometimes compounded with attached as well as casted shadows.

The diverse nature of surface materials of real-world objects manifests a wide range of specularities on the surface, impeding traditional photometric stereo methods (Wu et al., 2010; Ikehata et al., 2012; Mukaigawa et al., 2007; Wu & Tang, 2010). With the recent advent of deep learning techniques, tremendous advancements have been made in many computer vision problems, and there is no exception for photometric stereo (Santo et al., 2017; Ikehata, 2018; Li et al., 2019; Chen et al., 2020; Yao et al., 2020). Current existing deep PS methods often handle specularities in an implicit manner in the sense that an end-to-end supervised training process is adopted, ignoring the underlying physical principle of photometric stereo. The lack of interpretability of deep learning methods also prevents exploiting the interactions between specularities and surface normals. Moreover, shadows commonly appear in non-convex objects occluding part of the object surface, hindering photometric stereo reconstruction. Previous attempts were made to tackle photometric stereo in the presence of shadows, often under the restrictive Lambertian assumption (Chandraker et al., 2007). The problem becomes much complicated if both specularities and shadows appear simultaneously on the surface.

In the regime of deep photometric stereo, despite various data augmentation strategies (Santo et al., 2017; Li et al., 2019) were proposed to mitigate issues associated with shadows, they fall short in terms of overlooking how shadows occur on the surface. The lack of large-scale real-world training data also hinders the progress of data-driven deep photometric stereo. Moreover, obtaining ground-

truth 3D shapes and ground-truth svBRDFs in a real-world lab setting has been both laborious and expensive, preventing a wider up-taking of supervised learning approaches for non-Lambertian photometric stereo.

In this paper, we propose an unsupervised deep photometric stereo method that overcomes many of the aforementioned issues. Our framework takes the image coordinate corresponding to a surface point as the input, and directly outputs the surface normal, BRDF parameters (*i.e*. diffuse albedo and specular parameters), and depth at that surface point. To account for the different types of specularities in the real-world, our method learns a series of specular BRDF basis functions. With these BRDF bases, we parameterize the BRDF and fit the specular highlights to find the accurate surface normal. Furthermore, our framework explicitly parameterizes the shadowed regions by tracing through the estimated depth map. These shadowed regions are excluded from computation in order to avoid possible rendering artifacts. Following the inverse graphics rendering idea, we use the estimated surface properties and specular bases to re-render the pixel intensities of the surface point under different light directions. Our framework is optimized by minimizing the difference between the reconstructed and observed images during the inference time. Hence we do not need any ground truth surface normals nor pre-training. Our method outperforms both the supervised and self-supervised state-of-the-art methods on the challenging real-world dataset of DiLiGenT (Shi et al., 2018). Our framework is ten times faster than competing deep CNN based self-supervised photometric stereo methods (Taniai & Maehara, 2018; Kaya et al., 2020).

2 Related Work

Conventional approaches: The photometric stereo is firstly introduced by Woodham (Woodham, 1980), which assumes the surface of the objects to be Lambertian and convex to avoid the specular effects and shadows. This problem can therefore be solved in a closed-form manner by least-squares. The above strict assumptions were gradually liberalized by later studies (Wu et al., 2010; Ikehata et al., 2012; Mukaigawa et al., 2007; Wu & Tang, 2010; Quéau et al., 2017). These methods can tolerate the existence of non-Lambertian effects by treating the specularities and cast shadows on the object as outliers. However, they may also erase other clues specularities can bring.

Learning-based methods with ground truth surface normal at supervision: With the progress of deep learning in many of the computer vision areas, the learning-based methods are the ones that have achieved the best performance in photometric stereo recently (Santo et al., 2017; Ikehata, 2018; Li et al., 2019; Chen et al., 2020; Yao et al., 2020; Wang et al., 2020; Zheng et al., 2019). Santo et al. (2017) proposed the first network-based method, which per-pixelly estimates the normal by taking observed pixels in a pre-defined order. Chen et al. (2018; 2020) proposed a feature-extractor and features-pooling strategy to obtain the spatial information for photometric stereo. Recently, more works (Yao et al., 2020; Wang et al., 2020) exploited the local and global photometric clues for this problem. These learning-based methods require a large amount of data with ground truth surface normal at the training stage. The synthesized data with some augmentation strategies are commonly used as collecting a large-scale real-world dataset is exceptionally expansive and impractical.

Self-supervised methods: In contrast to the above-mentioned learning-based methods method, self-supervised methods do not require ground truth normal at supervision. Instead, the network is optimized by minimizing the difference between the reconstructed images and observed images. Taniai & Maehara (2018) proposed a self-supervised network that takes the whole set of images at the input, directly output the surface normal, and aiming to reconstruct the observed images. Their network structure is further expanded by Kaya et al. (2020) to deal with interreflection in the context of uncalibrated photometric stereo. Both of them implicitly encode specular components as features for the network and fail to consider shadows in the rendering equation.

Neural radiance fields: Recently, neural radiance fields introduced by Mildenhall et al. (2020) is widely adopt in many computer vision areas. Wizadwongsa et al. (2021) improve the novel view reconstruction results on view-dependent effects by using multiplane image and neural basis functions to represent the scene. Many works also extend the neural radiance fields to recover both the shapes and materials of the object (Boss et al., 2020; Zhang et al., 2021b;a; Srinivasan et al., 2021). These works are solving multi-view reconstruction problems. They generally assume the input being images of an object captured from multiple viewpoints under fix illumination. In

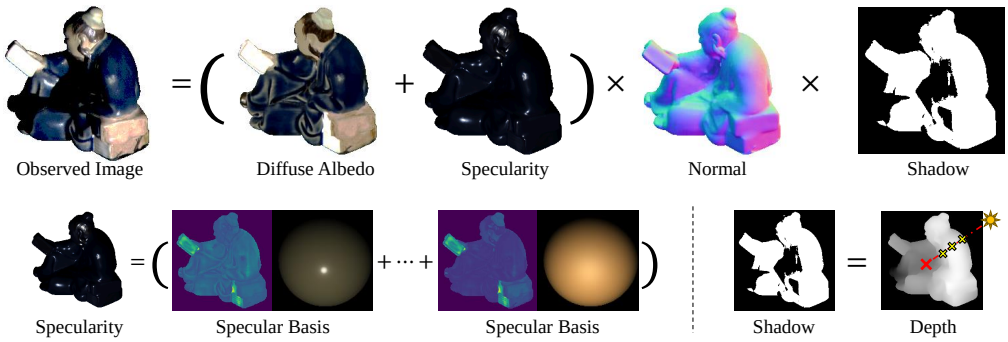


Figure 1: We propose an inverse rendering framework that estimates the surface normal, diffuse albedo, specularity, and shadow of an object. Our method learns the specular basis to fit the observed specularities accurately and gives clues for normal estimation. We also explicitly parameterize the shadows based on the estimated depth, alleviating artifacts on these shadows.

contrast, the photometric stereo problem we are focusing in this paper assumes multiple images taking from the same viewpoint, but with different illuminations.

3 Proposed Method

As shown in Fig. 1, our framework aims at inverse rendering the object by decoupling the surface into normal, diffuse albedo, specularity, and shadow. We model the specularity by learning the underlying specular bases. Our method estimates the depth by querying the relative depth of the surface points. In the following subsections, we illustrate the details of each module in our framework.

3.1 Rendering Equation

Following the conventional calibrated photometric stereo problem, we assume that the light source is in distance over the images with known light direction $\boldsymbol{l} = [l_x, l_y, l_z]^T \in \mathcal{S}^2$ (the space of 3-dimensional unit vectors) and light intensity $L_i \in \mathbb{R}_+$. And the camera to be in orthographic position, hence, viewing direction $\boldsymbol{v} = [0, 0, -1]^T \in \mathcal{S}^2$. For simplicity, without any loss of generality, we omit the light intensity L_i in the following formulations by dividing the observations (*i.e.* images I_i) with the corresponding lighting intensities, $I = I_i/L_i$. We also assume that there are no inter-reflections between the surfaces so that the point light source is the only light source to illuminate the target object.

Given a light source from the direction $\boldsymbol{l}$ illuminates a surface point with surface normal $\boldsymbol{n} \in \mathcal{S}^2$. The observation I viewing from direction $\boldsymbol{v}$ can be written as

$$I = s\rho(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}) \max(\boldsymbol{l}^T \boldsymbol{n}, 0), \tag{1}$$

where $s \in \{0, 1\}$ is a binary variable with a value of 0 at shadows, and 1 otherwise; $\rho(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n})$ represents the BRDF of the surface point, which is a function of the light, view direction, and the surface normal; $\max(\boldsymbol{l}^T \boldsymbol{n}, 0)$ is the shading component.

3.2 Material Modeling

The Lambertian surface assumes the BRDF $\rho(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}) = \rho_\mathrm{d}$ is always a positive constant. This unrealistic assumption fails to account for those materials with high specular effects. It can be beneficial to model the specular part in BRDF and leverage its information for photometric stereo. In order to take both the diffuse and specular effects into account, here we choose a more realistic way to model the BRDF, *i.e.* the microfacet BRDF models (Torrance & Sparrow, 1967; Walter et al., 2007), where the BRDF is separated into the diffuse and specular components

$$\rho(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}) = \rho_\mathrm{d} + \rho_\mathrm{s}(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}). \tag{2}$$

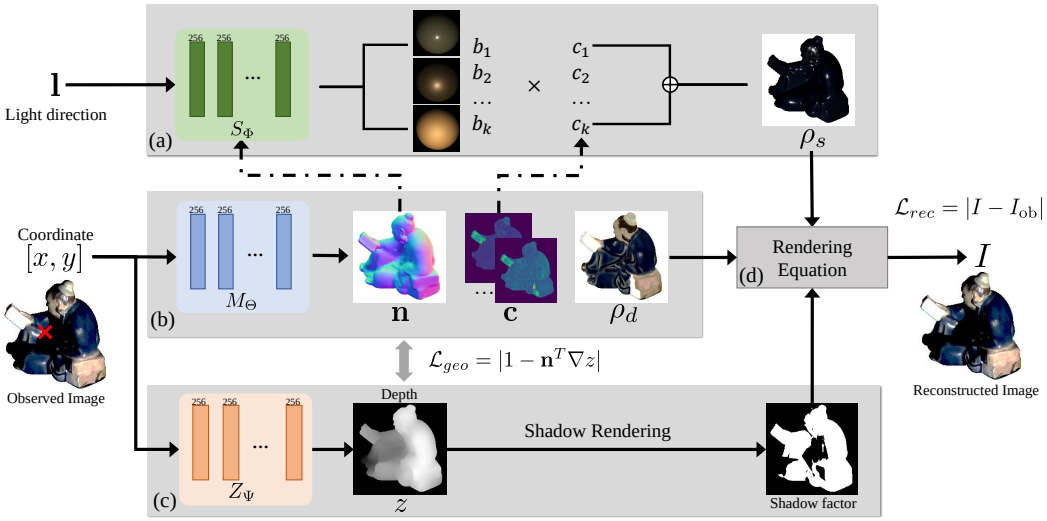


Figure 2: The four modules of our MLP-based deep photometric stereo framework: (a) specularity modeling S_Φ (see Sec. 3.2) fits a suitable set of suitable BRDF bases to the target specularities; (b) surface modeling M_Θ (see Sec. 3.3) estimates the surface normal, as well as parameters of the BRDF given the image coordinates as input; (c) Z_Ψ estimates a dense depth map, which enables the shadow rendering (see Sec. 3.4) by checking the visibility of the light source at each surface point; and (d) the rendering equation (see Sec. 3.1). All MLPs are optimized in a self-supervised manner by minimizing the reconstruction error between reconstructed and observed images.

Specularity modeling. Previous deep-learning-based approaches implicitly handle the specularity on images by feeding them as features into their neural network (Taniai & Maehara, 2018; Kaya et al., 2020), or processed by max-pooling (Chen et al., 2018; 2020). However, as the specularities, at the core, are reflections on the surface, explicitly model these effects by using clues from physical reflection constraints will certainly bring merits to the photometric stereo problem.

To relieve the burden of fitting such a specular BRDF, we need to introduce some reasonable and realistic assumptions. Recalling that the BRDF can be converted to a half-vector $\boldsymbol{h}$ based function with only four parameters (Rusinkiewicz, 1998), we assume that our specular BRDF is isotropic and is only the function of half-vector $\boldsymbol{h}$ and surface normal $\boldsymbol{n}$. This assumption omits the Fresnel reflection coefficient and the geometric attenuation, which only has limited effects at grazing angles (Burley & Studios, 2012). Besides, observing the fact that many surface points in the real-world object are similar, if not identical, in the material. We further assume that the specular BRDF $\rho_{\text{s}}(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n})$ at each surface point lies on a non-negative linear combination of the atoms of specular basis. Similar approaches for simplifying the BRDF model to be the combination of different bases were also used in (Matusik et al., 2003; Hui & Sankaranarayanan, 2017). The specular BRDF can then be written as

$$\rho_{\text{s}}(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}) = \mathbf{c}^T \, D(\boldsymbol{h}, \boldsymbol{n}), \quad \boldsymbol{h} = \frac{\boldsymbol{l} + \boldsymbol{v}}{||\boldsymbol{l} + \boldsymbol{v}||}, \tag{3}$$

where $\boldsymbol{h}$ is the half-vector between lighting and viewing direction; and $D(\boldsymbol{h}, \boldsymbol{n}) = [b_1, b_2, \cdots, b_k]^T$ is the underlying specular basis of the target object; $[c_1, c_2, \cdots, c_k]^T := \mathbf{c} \in \mathbb{R}_+^k$ represent the weights of each specular basis; k is the number of different bases. We assume that $\mathbf{c}$ is an element-wise non-negative vector, suggesting that the surface reflectance is represented by positive combination of a small number of basis materials.

We have studied two different ways to model non-Lambertian BRDF basis: one uses the MLP to learn possible specular BRDF basis of the target object, the other is based on Spherical Gaussian.

MLP Basis. An MLP can represent the specular basis by

$$D(\boldsymbol{h}, \boldsymbol{n}) = S_\Phi(\boldsymbol{h}, \boldsymbol{n}), \tag{4}$$

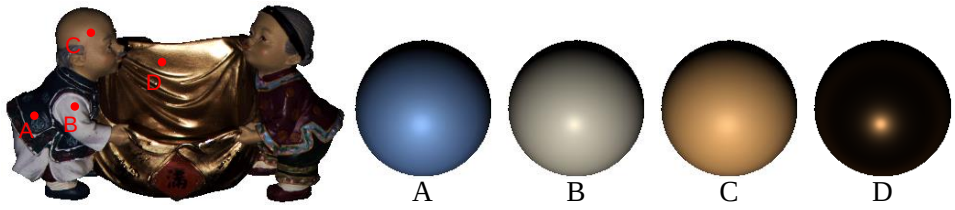


Figure 3: **Visualization on the estimated svBRDFs.** We select four different surface points on the object "Harvest" and showcase our estimated BRDF spheres on the right. The results demonstrate that our model can recover the metallic and diffuse materials. We scale up the observed images and normalize the BRDF spheres for better visualization.

The network $S_\Phi(\boldsymbol{h}, \boldsymbol{n})$ only takes $\boldsymbol{h}, \boldsymbol{n}$ at the input, outputs the different specular basis in form of $[b_1, b_2, \cdots, b_k]^T$, as shown in Fig. 2. Φ are its weights that can be optimized. In Fig. 3, we showcase the estimated svBRDFs by our MLP specular model.

Spherical Gaussian Basis. Alternatively, we can also use the Spherical Gaussians (Wang et al., 2009) to represent the specular basis by

$$D(\boldsymbol{h}, \boldsymbol{n}) = G(\boldsymbol{h}, \boldsymbol{n}; \lambda) = [e^{\lambda_1(\boldsymbol{h}^T\boldsymbol{n}-1)}, \cdots, e^{\lambda_k(\boldsymbol{h}^T\boldsymbol{n}-1)}]^T \tag{5}$$

$\lambda_i \in \mathbb{R}_+$ represents the sharpness of the specular peak. By choosing $\lambda = [\lambda_1, \cdots, \lambda_k]^T \in \mathbb{R}_+^k$ with different values, we can create different types of specular basis functions of different sharpness.

In experiments, we found that the MLP basis has a slight advantage on representing metallic objects over Spherical Gaussian. Because Spherical Gaussian has difficulties in representing high-peak and long-tail specularites. On the other hand, due to having much fewer parameters, Spherical Gaussian is more computationally efficient than the MLP model.

3.3 Surface Modeling

We model the surface normal, diffuse, and specular basis coefficients of an object by an MLP M_Θ. It takes the image coordinates of the pixels $\boldsymbol{x} = [x, y]^T \in \mathbb{R}^2$ as input. The output is the corresponding surface normal $\boldsymbol{n}$, diffuse albedo ρ_d, and the coefficients $\mathbf{c}$ of the specular component at each coordinate $\boldsymbol{x}$.

$$\boldsymbol{n}, \rho_\text{d}, \mathbf{c} = M_\Theta(\boldsymbol{x}), \tag{6}$$

where $\mathbf{c}$ represents the coefficients that can be used to reconstruct the specular component ρ_s in Sec. 3.2; and Θ is the weights of this MLP that can be optimized.

We use the similar MLP architecture and positional encoding strategy from NeRF (Mildenhall et al., 2020) to build our network, and the embedding in input coordinates $\boldsymbol{x}$. The difference is that while NeRF also takes different viewing directions as input to model the view-dependent effects of the objects' appearance, our M_Θ network only estimates the "static" properties of the target object. Instead, we cover the "light-dependent" variance of the object by specularity modeling. Our design will encourage the network to correctly decompose surface normal and material property of the object.

3.4 Shadow Handling

We now look at the shadow factor s in the image rendering Eq. (1). Due to the rugged surface of the objects in the world, shadows may appear at the reflecting surface. As shown in Fig. 4, shadow occurs when the object itself occludes the surface. Rendering of the shadowed region relies on the relative geometry and depth of the object with respect to the light directions. Hence, we introduce a depth MLP Z_Ψ to model the object's depth value $z \in \mathbb{R}$ between the object surface points to the camera. The depth MLP takes images coordinates as input, outputs the corresponding depth value of the given coordinates $z = Z_\Psi(\boldsymbol{x})$.

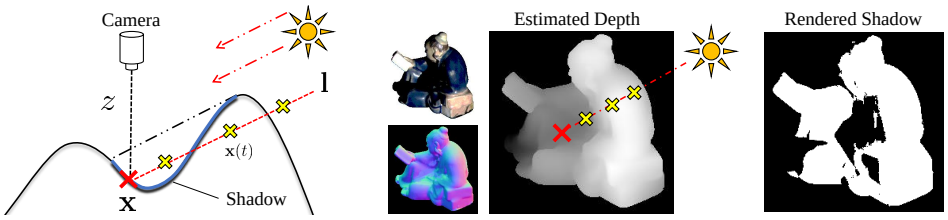


Figure 4: Shadow parameterization and rendering. As shown in the left figure, shadows are caused by self-occlusion. To determine whether a surface point $\boldsymbol{x}$ falls into the shadow region, we trace the point to the light source and sample multiple points $\boldsymbol{x}(t)$ along this ray. Given the light direction $\boldsymbol{l}$ and the estimated depth map $Z_\Psi(\boldsymbol{x})$, we can query the depth and compare the values to effectively parameterize and render the shadow by Eq. (7).

To examine whether the object occludes the light source and hence causing the shadow, we can draw a line from the surface point $\boldsymbol{x}$ toward the light source. Denote this line in the world coordinates as $\mathbf{L} = \mathbf{X} - t\boldsymbol{l}$, where $t \in (0, +\infty)$; the $\mathbf{X} = [x, y, z]$ represent the surface points with its depth value z given by $Z_\Psi(\boldsymbol{x})$. We can further simplify the equation by using the function L_z to denote the z-axis value of $\mathbf{L}$. Now, by traveling along the light direction, *i.e.* $t \in (0, +\infty)$, we can compute the shadow factor by

$$s = \text{step}\left(\min_{\boldsymbol{x}(t)} \left(Z_\Psi(\boldsymbol{x}(t)) - L_z(\boldsymbol{x}(t))\right)\right), \quad \boldsymbol{x}(t) = \boldsymbol{x} - t\boldsymbol{l}', \tag{7}$$

where the $\text{step}(\cdot)$ denote the Heaviside step function, which outputs 1 if input is positive, and 0 otherwise; $\boldsymbol{l}' = [l_x, l_y]^T$ is the projection of light direction $\boldsymbol{l}$ at xy-plane. In implementation, we set the step size for shadow rendering to be 32 (with logspace intervals).

4 Implementation

We use the positional encoding strategy to encode the input before inputting them into the MLP (c.f. Mildenhall et al. (2020)). For surface modeling net M_Θ, we encode the input with 10 levels of Fourier functions, the network M_Θ uses 12 fully-connected ReLU layers with 256 channels. The surface normal $\boldsymbol{n}$ is output at 8-th layer while the BRDF parameters are output at the last layer. We also use 10 encoding functions to embed the input of depth net Z_Ψ, which has 8 fully-connected ReLU layers with 256 channels. For the specular MLP S_Φ, we use only 3 encoding functions to embed the input. The network S_Φ consists of 3 fully-connected ReLU layers with 64 channels. Overall, the three MLP-networks are rather lightweight (*i.e.* small footprint) with total combined parameters of merely 1.1M. Please refer to the appendix for more implementation details.

Reconstruction loss The reconstruction loss is defined as mean absolute errors between the observed intensity I_{ob} and reconstructed intensity:

$$\mathcal{L}_{rec} = \sum_{\text{all pixels}} |I - I_{\text{ob}}|. \tag{8}$$

Geometry Constraint Note that the Heaviside step function $\text{step}(\cdot)$ will cutoff the gradients from $\mathcal{L}_{rec}$ to the network parameters of Z_Ψ. Hence we introduce a geometry constraint between the estimated surface normal $\boldsymbol{n}$ and depth network Z_Ψ as below

$$\mathcal{L}_{geo} = \sum_{\text{all pixels}} (1 - \boldsymbol{n}^T \nabla Z_\Psi). \tag{9}$$

Besides, in the early stage of optimizing the network Z_Ψ, its output depth map cannot reason a stable and reasonable shadow mask. We introduce shadow guidance s_g to replace the rendered shadow factor at Eq. (7) at the beginning of optimization. Assume that observation under n different light direction is $[I_1, I_2, ..., I_n]$. We then set a threshold as $0.1\lambda_m$, where $\lambda_m = \frac{1}{n}\sum I_i$ is the mean intensity value of the different observations. Those pixel intensities that are smaller than the threshold will be considered as shadowed, hence with shadow factor $s_g = 0$. We use Eq. (7) for shadow rendering once the depth network Z_Ψ is stable.

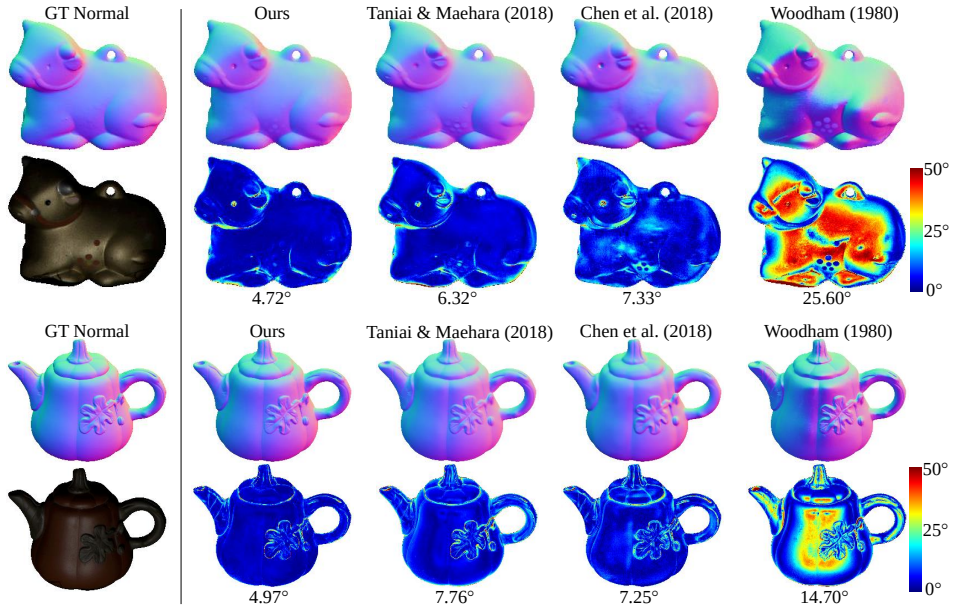


Figure 5: Qualitative results on "Cow" and "Pot2". For each object, the odd numbered rows show the observed image and estimated normal by different methods; the even numbered rows show the angular (normal) error in degrees by different methods.

Table 1: Quantitative comparison on the DiLiGenT dataset. The metric here is mean angular error (MAE); the lower MAE is preferred.

GT normal	Methods	Ball	Bear	Buddha	Cat	Cow	Goblet	Harvest	Pot1	Pot2	Reading	Avg.
No	Ours	2.43	**3.64**	8.04	4.86	**4.72**	**6.68**	14.90	5.99	**4.97**	**8.75**	**6.50**
No	Taniai & Maehara (2018)	**1.47**	5.79	10.36	5.44	6.32	11.47	22.59	6.09	7.76	11.03	8.83
No	Woodham (1980)	4.10	8.40	14.90	8.40	25.60	18.50	30.60	8.90	14.70	19.80	15.40
Yes	Wang et al. (2020)	1.78	4.12	**6.09**	4.66	6.33	7.22	**13.34**	6.46	6.45	10.05	6.65
Yes	Ikehata (2018)	2.20	4.10	7.90	**4.60**	8.00	7.30	14.00	**5.40**	6.00	12.60	7.20
Yes	Yao et al. (2020)	2.92	5.07	7.77	5.42	6.14	9.00	15.14	6.04	7.01	13.58	7.81
Yes	Chen et al. (2018)	2.82	5.15	7.91	5.98	7.33	8.60	15.85	6.85	7.25	13.33	8.08

Smoothness constraint As the albedo and surface normal of real-world objects usually present a piece-wise smooth pattern, we introduce a total variation loss to guide the network in the early stage.

$$\mathcal{L}_{tv} = V_{l_1}(\rho_{\mathrm{d}}, \mathbf{c}) + V_{l_2}(\boldsymbol{n}), \tag{10}$$

where V_{l_1} represents the total variation function with absolute loss and V_{l_2} with square loss.

To sum up, we optimize the parameters of the MLPs M_Θ, S_Φ, Z_Ψ by minimizing the following loss function: $\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{geo} + \beta\mathcal{L}_{tv}$, where β is the hyper-parameter controlling the total variation loss. We set it as $\beta = 0.01$; and it will then be set to 0 after the first 2400 iterations.

5 Experiments

In this section, we evaluate our method and its variants on the challenging real-world dataset DiLiGenT (Shi et al., 2018). We used all the $n = 96$ images under different light directions for optimizing the network, except the object "Bear". We discard the first 20 images of "Bear", as they are found to be over-saturated in previous work (Ikehata, 2018). The batch size is set as 8 images per batch. We iterate in total 6000 iterations when optimizing the network. We use Adam (Kingma & Ba, 2014) optimizer with a learning rate of 5×10^{-4} and other parameters at their default settings. Our method is implemented in PyTorch and is running on a RTX 3090 GPU. The inference (i.e. training) time in the 10 objects of DiLiGenT dataset range from 3 min to 9 min, with an average of 6 min per object. In contrast, Taniai & Maehara (2018) and Kaya et al. (2020) took about an hour per object.

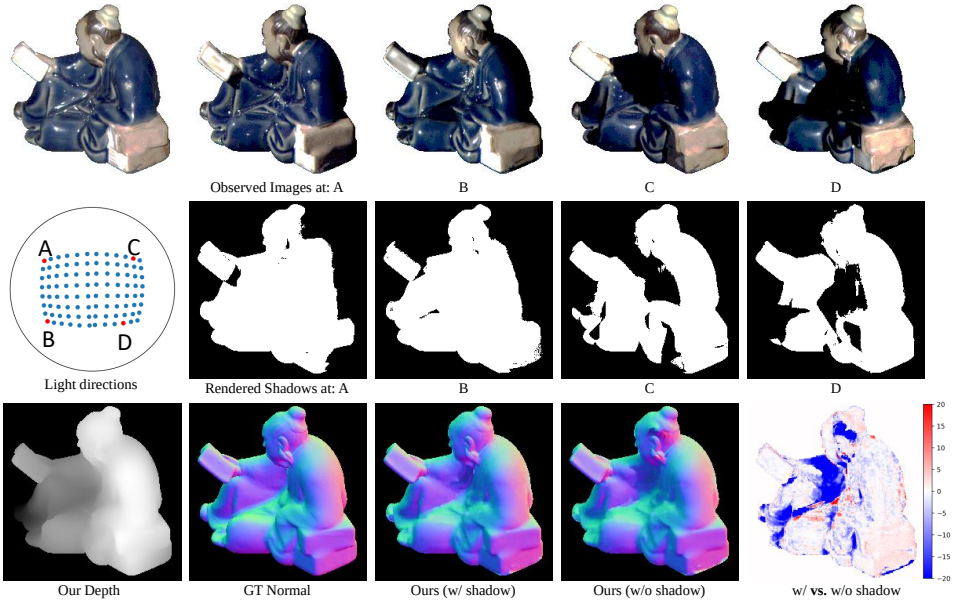


Figure 6: We select four different light directions. Their distribution is labeled as red points in the light distributions image in the second row. The first row shows the observed images under these four different light sources. The second row presents the results of our rendered shadow region under the corresponding illuminations. In the third row, we showcase the estimated depth, ground truth surface normal, estimated surface normal (with and without the shadow factor). In the right-most image on the third row, we also compare our estimated normal "w/ shadow" and "w/o shadow". The blue color in the comparison corresponds to the area where "w/ shadow" outperforms "w/o shadow".

5.1 Evaluation on real-world dataset

In Table 1, we present the quantitative comparison of our method against other methods on the DiLiGenT dataset. We use the mean angular error (MAE) as the metric in the paper. The lower MAE is preferred. We classify the previous methods into two categories: the supervised methods, which need ground truth surface normal at the training stage; and the self-supervised which does not need ground truth surface normal and directly estimates the normal at testing time. As reported in the Table 1, our method achieves the best performance over the other methods at average MAE errors. Thanks to our specularity modeling, our method shows its significant advantages on shiny objects like "Reading", "Cow" and "Goblet". We present the visualization of "Cow" and "Pot2" in Fig. 5. "Cow" is a typical metallic object with a high peak of specularities; while "Pot2" shows more broad and soft specular effects. Our method achieves the best performance in both two cases.

5.2 Evaluation of the specular models

Here, we compare our method with different choices of the specular model and different numbers of the basis functions. We experimentally find out that our method with MLP to model the specular basis will have advantages over metallic objects. The metallic material usually has high-peak and long-tail effects on the BRDF map. The Spherical Gaussian struggles to accurately recover these types of BRDF, while the MLP model can fit the metallic material very well. From the Table 2, both of the two modeling methods achieve the best performance when the number of bases is set to $k = 9$. Note that, despite the Spherical Gaussian shows slightly worse at performance, it requires less computational resources; and it is twice faster than using the MLP.

5.3 Shadow handling

To show the efficacy of our shadow handling mechanism, we conduct ablation study by removing the shadow rendering module, denote as "w/o shadow". Quantitative comparisons are shown in

Table 2: Quantitative comparison on the different specular models and number of basis. The metric here is MAE; lower is preferred.

Specular model \ #Basis	2	4	6	9	12
MLP	6.80	6.77	6.65	**6.50**	7.39
SG	7.10	6.89	6.80	<u>6.79</u>	7.04

Table 3: Evaluations of the different variants of the proposed method. The second row is the method without the shadow factor; the third row is without using the early stage smoothness constraint. The metric here is MAE; lower is preferred.

Methods	Ball	Bear	Buddha	Cat	Cow	Goblet	Harvest	Pot1	Pot2	Reading	Avg.
Proposed	2.43	**3.64**	**8.04**	**4.86**	**4.72**	**6.68**	**14.90**	**5.99**	**4.97**	**8.75**	**6.50**
w/o shadow	**2.13**	4.29	11.09	6.81	5.69	8.30	17.88	7.79	7.80	12.68	8.44
w/o $\mathcal{L}_{tv}$	2.44	3.66	8.56	4.93	5.27	6.77	21.67	6.73	6.88	9.19	7.61

Table 3, where one can see that the mean angular error on all objects increases 1.96 degrees. Notably, the performance degradation is majorly caused by objects "Buddha", "Harvest" and "Reading". This is as expected, because these objects have rather complex (concave) surface geometry, more susceptible to cast shadows. Our proposed shadow handling method attends to these shadowed regions better, achieving high recovery accuracy. In Fig. 6, we give the visualization of the effects of our shadowing module on the object "Reading". Observing the image and its ground truth normal of this object, we can see that "Reading" is a highly non-convex object with many specularities and shadows. The shadowed region is especially big when the light comes from the right direction, as shown in the lighting direction C and D in the figure. Our render shadows under these lighting directions, despite some minor errors, accurately predicting the shadowed regions. The error map shown at the right-most of the third row in Fig. 6 corresponds to the difference between the MAE yielded by our proposed model ("w/ shadow") and its no-shadow-variant ("w/o shadow"). The negative areas, *i.e.*, blue regions in the error map, are those where our proposed model outperforms the alternative. The full model performs better in the region where the shadows are evident.

6 Discussions and Conclusions

In this paper, we have proposed an MLP-based approach for non-Lambertian photometric stereo. The key novelty of our method is the neural parameterizations of spatially-varying surface reflectances (svBRDF), and of surface geometry. By leveraging the physical principle of inverse rendering, we explicitly tackle the specularities and cast shadows in the photometric stereo setting. Despite being an unsupervised method, our method outperforms existing state-of-the-art supervised methods on real-world datasets. Our method is inspired by NeRF (Mildenhall et al., 2020), which also uses a coordinate-based MLP to model the mapping from 3D coordinates to surface materials and appearance. In contrast, we factorize the image appearance into multiple components: normal, diffuse, specular, and shadows. The fitting on these physical-based rendering factors restores the object's surface properties faithfully. Our method also relates to Taniai & Maehara (2018); Kaya et al. (2020), which aim at optimizing a CNN-based inverse rendering architectures. Our MLP-based framework is significantly faster than those CNN-based methods. Besides, we explicitly parameterize diffuse, specularities, and shadows to ensure the inverse rendering follows a physically meaningful and explainable manner.

Limitations and future work: We believe that solving the photometric stereo without known light sources (i.e. uncalibrated photometric stereo) would be an interesting future work for us. Our estimation of depth is learned from the surface normals by the geometry constraints. Hence, it is sensitive to the accuracy of normal and surface discontinuities. Introducing more constraints for accurate depth estimation would certainly help this problem. Another challenging topic is to include inter-reflection and global illumination in the rendering equation. Finding a model to trace the rays bouncing between surfaces is also an interesting future direction.

7 ETHICS STATEMENT

As discussed by Yao *et al*. (Yao et al., 2020), with the advancement of photometric stereo, the 3D shape of a person's face can be easily captured accurately even by a layman. The inverse rendering technique allows the user to alter the shape and appearance of an individual's face. The acquisition and alteration of such personal information, if without their consent, may lead to privacy and security breaching. Care must be taken to mitigate the potential risk of abusing this technique.

8 REPRODUCIBILITY STATEMENT

Implementation details The detail design of our network architecture can be found in main paper Section 5 and the appendix Section A. All the training details are provided in Section 5 and Section 5.2. We will release the code once the paper is public available.

Datasets used in the paper In our paper, we conduct the experiments in three public real-world datasets: DiLiGenT (Shi et al., 2018), Gourd&Apple dataset (Alldrin et al., 2008), and Light Stage Data Gallery (Chabert et al., 2006). Results for Gourd&Apple, and Light Stage Data Gallery are shown in the appendix. All these three datasets were acquired in the dark room. All three datasets provide calibrated light directions and light intensities. DiLiGenT contains 10 objects; each object has 96 images captured under different lighting conditions; ground truth surface normal is captured by a high-end laser scanner. Gourd&Apple contains 3 objects; each object has around 100 images captured under different lighting conditions; no ground truth surface normal is provided. Light Stage Data Gallery contains 5 objects; each object has around 133 images available for photometric stereo; and no ground truth surface normal is provided.

REFERENCES

Neil Alldrin, Todd Zickler, and David Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2008.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. *arXiv preprint arXiv:2012.03918*, 2020.

Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pp. 1–7. vol. 2012, 2012.

Charles-Félix Chabert, Per Einarsson, Andrew Jones, Bruce Lamond, Wan-Chun Ma, Sebastian Sylwan, Tim Hawkins, and Paul Debevec. Relighting human locomotion with flowed reflectance fields. In *ACM SIGGRAPH 2006 Sketches*, pp. 76–es. 2006.

Manmohan Chandraker, Sameer Agarwal, and David Kriegman. Shadowcuts: Photometric stereo with shadows. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2007.

Guanying Chen, Kai Han, and Kwan-Yee K Wong. Ps-fcn: A flexible learning framework for photometric stereo. In *European Conference on Computer Vision*, pp. 3–19. Springer, 2018.

Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee Kenneth Wong. Deep photometric stereo for non-lambertian surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Zhuo Hui and Aswin C Sankaranarayanan. Shape and spatially-varying reflectance estimation from virtual exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(10): 2060–2073, 2017.

Satoshi Ikehata. Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In *European Conference on Computer Vision*, pp. 3–19. Springer, 2018.

Satoshi Ikehata, David Wipf, Yasuyuki Matsushita, and Kiyoharu Aizawa. Robust photometric stereo using sparse regression. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 318–325. IEEE, 2012.

Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncalibrated neural inverse rendering for photometric stereo of general surfaces. *arXiv preprint arXiv:2012.06777*, 2020.

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

Junxuan Li, Antonio Robles-Kelly, Shaodi You, and Yasuyuki Matsushita. Learning to minify photometric stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7568–7576, 2019.

Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003.

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 *European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

Yasuhiro Mukaigawa, Yasunori Ishii, and Takeshi Shakunaga. Analysis of photometric factors based on photometric linearization. *JOSA A*, 24(10):3326–3334, 2007.

Romain Pacanowski, Oliver Salazar Celis, Christophe Schlick, Xavier Granier, Pierre Poulin, and Annie Cuyt. Rational brdf. *IEEE transactions on visualization and computer graphics*, 18(11): 1824–1835, 2012.

Yvain Quéau, Tao Wu, François Lauze, Jean-Denis Durou, and Daniel Cremers. A non-convex variational approach to photometric stereo under inaccurate lighting. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 350–359. IEEE, 2017.

Szymon M Rusinkiewicz. A new change of variables for efficient brdf representation. In *Eurographics Workshop on Rendering Techniques*, pp. 11–22. Springer, 1998.

Hiroaki Santo, Masaki Samejima, Yusuke Sugano, Boxin Shi, and Yasuyuki Matsushita. Deep photometric stereo network. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pp. 501–509. IEEE, 2017.

Boxin Shi, Zhipeng Mo, Zhe Wu, Dinglong Duan, Sai Kit Yeung, and Ping Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7495–7504, 2021.

Tatsunori Taniai and Takanori Maehara. Neural inverse rendering for general reflectance photometric stereo. In *International Conference on Machine Learning*, pp. 4864–4873, 2018.

Kenneth E Torrance and Ephraim M Sparrow. Theory for off-specular reflection from roughened surfaces. *Josa*, 57(9):1105–1114, 1967.

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. *Rendering techniques*, 2007:18th, 2007.

Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Transactions on Graphics (TOG)*, 28(5):1–10, 2009.

Xi Wang, Zhenxiong Jian, and Mingjun Ren. Non-lambertian photometric stereo network based on inverse reflectance model with collocated light. *IEEE Transactions on Image Processing*, 29: 6032–6042, 2020.

Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8534–8543, 2021.

Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139, 1980.

Lun Wu, Arvind Ganesh, Boxin Shi, Yasuyuki Matsushita, Yongtian Wang, and Yi Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *Asian Conference on Computer Vision*, pp. 703–717. Springer, 2010.

Tai-Pang Wu and Chi-Keung Tang. Photometric stereo via expectation maximization. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):546–560, 2010.

Zhuokun Yao, Kun Li, Ying Fu, Haofeng Hu, and Boxin Shi. Gps-net: Graph-based photometric stereo network. *Advances in Neural Information Processing Systems*, 33, 2020.

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*, pp. 5453–5462, 2021a.

Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *arXiv preprint arXiv:2106.01970*, 2021b.

Qian Zheng, Yiming Jia, Boxin Shi, Xudong Jiang, Ling-Yu Duan, and Alex C Kot. Spline-net: Sparse photometric stereo through lighting interpolation and normal estimation networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8549–8558, 2019.

APPENDIX

A IMPLEMENTATION DETAILS

Positional encoding of input of M_Θ and Z_Ψ The inputs of surface modeling M_Θ and depth modeling Z_Ψ are the pixel coordinates $\boldsymbol{x}$. We adopt the positional encoding strategy (Mildenhall et al., 2020) to embed the input coordinates $\boldsymbol{x} \in \mathbb{R}^2$ into a higher space $\boldsymbol{x} \in \mathbb{R}^{4m}$:

$$\gamma(\xi) = (\sin(2^0\pi\xi), \cos(2^0\pi\xi), \cdots, \sin(2^{m-1}\pi\xi), \cos(2^{m-1}\pi\xi)). \tag{11}$$

In practice, we first normalized the coordinates to range $(-1, 1)$, then apply the above encoding function with $m = 10$ to each of the two coordinate values in $\boldsymbol{x}$. Then, we concatenate the coordinate and its embeddings as $(\boldsymbol{x}, \gamma(\boldsymbol{x}))$ to be the input of the two MLPs.

Positional encoding of input of S_Φ Rusinkiewicz (Rusinkiewicz, 1998) reparameterized the BRDF as a function of the half-vector $\boldsymbol{h}$ (*i.e.* the half-vector between lighting and viewing direction). This half-vector-based parameterization is further evaluated and discussed by (Pacanowski et al., 2012; Burley & Studios, 2012), where they found that a simplified isotropic BRDF can be modeled in two parameters (θ_h, θ_d), where $\theta_h = \arccos(\boldsymbol{n}^T\boldsymbol{h}), \theta_d = \arccos(\boldsymbol{v}^T\boldsymbol{h})$. In our method, we take the cosine value of these two variables as the input:

$$\mathbf{p} = (\boldsymbol{n}^T\boldsymbol{h}, \boldsymbol{v}^T\boldsymbol{h}). \tag{12}$$

Then, the input $\mathbf{p}$ is further encoded by $\gamma(\mathbf{p})$ with $m = 3$. Likewise, we concatenate $\mathbf{p}$ and its embeddings as $(\mathbf{p}, \gamma(\mathbf{p}))$ to be the input of the specularity basis modeling S_Φ.

Network architecture In Fig. 7, we show the detail structure of our three MLPs : (a) specularity basis modeling S_Φ; (b) surface modeling M_Θ, (c) depth modeling Z_Ψ. The design of these structures are inspired by a recent work NeRF (Mildenhall et al., 2020).

B COMPARISON ON SHADOW GUIDANCE VERSUS SHADOW RENDERING

Here we conduct a comparative experiment on using only the shadow guidance versus using only the shadow rendering during the training. The results are shown in Table 4. We can see that the full model achieve the best performance among them.

Table 4: Quantitative comparison on using only the shadow guidance versus using only the shadow rendering during the training. The metric here is MAE; lower is preferred.

Methods	Ball	Bear	Buddha	Cat	Cow	Goblet	Harvest	Pot1	Pot2	Reading	Avg.
Full model	2.43	**3.64**	**8.04**	**4.86**	**4.72**	**6.68**	**14.90**	**5.99**	**4.97**	**8.75**	**6.50**
Only shadow guidance	**1.91**	3.77	**7.92**	5.86	**4.65**	7.11	15.11	6.20	5.43	**8.60**	6.66
Only shadow rendering	2.04	4.17	9.54	5.49	5.64	7.35	17.11	7.63	6.76	10.12	7.58

C COMPARISON ON DIFFERENT VARIANTS OF SURFACE MODELING

In this section, we try and compare several variants of our surface modeling network M_Θ. The original model we use in the paper is denoted as the baseline. The equation for baseline surface modeling (i.e. the one used in main paper) is

$$\boldsymbol{n}, \rho_\mathrm{d}, \mathbf{c} = M_\Theta(\boldsymbol{x}), \tag{13}$$

Surface Modeling Variant 1: directly output depth, diffuse albedo and specular weights rather than outputting normals.

$$z, \rho_\mathrm{d}, \mathbf{c} = M^1_\Theta(\boldsymbol{x}), \tag{14}$$

Surface Modeling Variant 2: output the normals and albedo by two networks separately.

$$\boldsymbol{n} = M^2_\Theta(\boldsymbol{x}), \tag{15}$$

$$\rho_\mathrm{d}, \mathbf{c} = M^3_\Theta(\boldsymbol{x}) \tag{16}$$

Table 5: Quantitative comparison on different variants of surface modeling. The metric here is MAE; lower is preferred. Below, we present the average MAE of ten objects in DiLiGenT.

Methods	Avg.
Baseline	**6.50**
Variant 1	7.56
Variant 2	6.64

Our experiments in Table 5 show that the baseline model (outputting normals, albedo and specular weights by a single network) has the best performance. Our analysis is that:

- Comparing to **Variant 1**: Because the surface normal is close related to the photometric appearance of an object. By directly outputting normal, our baseline network can achieve a lower photometric appearance (reconstruction) loss. If we directly output depth, we need to apply an additional step to compute the finite difference to get normal. Hence, directly outputting normal will help the training of the network to minimize the reconstruction loss.
- Comparing to **Variant 2**: Our baseline model will output the normals, albedo and specular weights in a single MLP, which encourages learning on shared features of surface properties. The normals, albedo and specular weights are all intrinsic properties of an object's surface. They are correlated with the image coordinates. Hence, outputting all these three values in a single network will encourage learning on their shared features.

D ADDITIONAL RESULTS

Additional Results on Normal Estimation In Fig. 8, we present the normal estimation results on three specular objects: "Goblet", "Reading" and "Harvest" from DiLiGenT (Shi et al., 2018). These results demonstrate that our method is taking advantage of the information that the specularities provide. Hence, we can estimate the normal accurately on specular regions.

Visual comparison on different choices on the specular model As shown in Fig. 9, both the MLP and the Spherical Gaussian models reconstruct the reflectance visually accurately. The MLP basis has a slight advantage on representing metallic objects over Spherical Gaussian. Because Spherical Gaussian has difficulties in representing high-peak and long-tail specularites.

Additional Results on svBRDF Estimation In Fig. 10, we showcase the estimated materials of different points on the object "Reading" by rendering the corresponding BRDF spheres.

Visualization on each terms of the rendering equation Recall the rendering equation (Eq.(1) in the main paper) is defined by

$$I = s\rho(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}) \max(\boldsymbol{l}^T \boldsymbol{n}, 0), \quad \rho(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{n}) = \rho_{\mathrm{d}} + \rho_{\mathrm{s}}, \tag{17}$$

where ρ_{d} is the diffuse albedo; ρ_{s} is the specularities; s is the shadows; $\max(\boldsymbol{l}^T \boldsymbol{n}, 0)$ is the shading term. In Fig. 11, we present the visualization of our estimation on these terms.

Additional Results on Shadows Estimation In Fig. 12, we showcase the estimated shadows and specularities under different light directions.

Additional Results on Other Real-world Dataset We also test our method on two other challenging real-world datasets: Gourd&Apple dataset (Alldrin et al., 2008) and Light Stage Data Gallery (Chabert et al., 2006), as shown in Fig. 13 and Fig. 14 separately. Both of these two datasets do not provide ground truth normal for evaluation. Hence, we provide the visualization of the estimated normal, diffuse albedo, and specular map on these datasets. Our method correctly recovers the shape and materials of different objects. It also demonstrates that our method is robust on different objects with different materials.

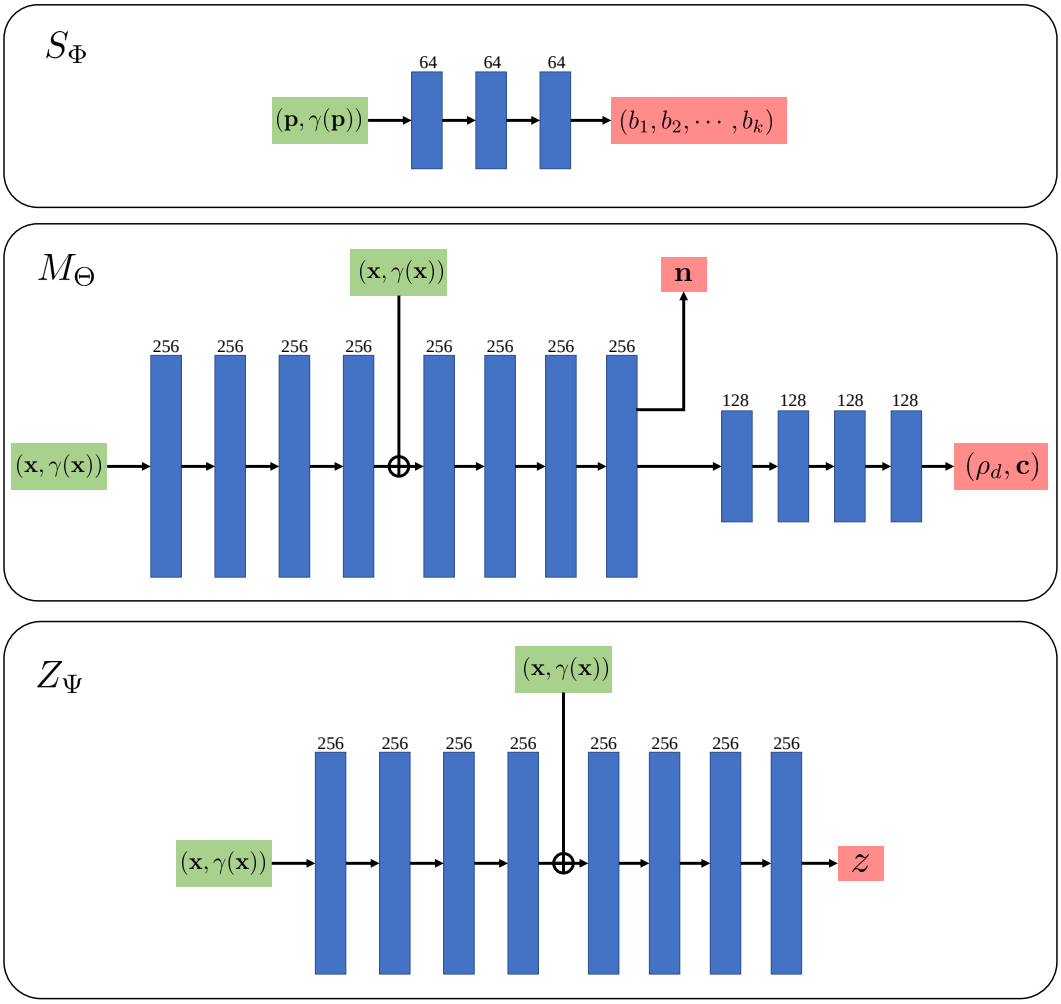


Figure 7: **Network architecture of our three MLPs:** S_Φ, M_Θ, Z_Ψ. In this figure, inputs of the network are shown in the green blocks; outputs are shown in the red blocks. The blue blocks represent the fully-connected layers with its size of the hidden channels stated on the top. All fully-connected layers are followed by a ReLU activation layer, except the output layers. The "$\oplus$" in the middle of the M_Θ, Z_Ψ network denotes the vector concatenation: we add a skip connection after the fourth layer of M_Θ, Z_Ψ, and concatenate its output features with the input.

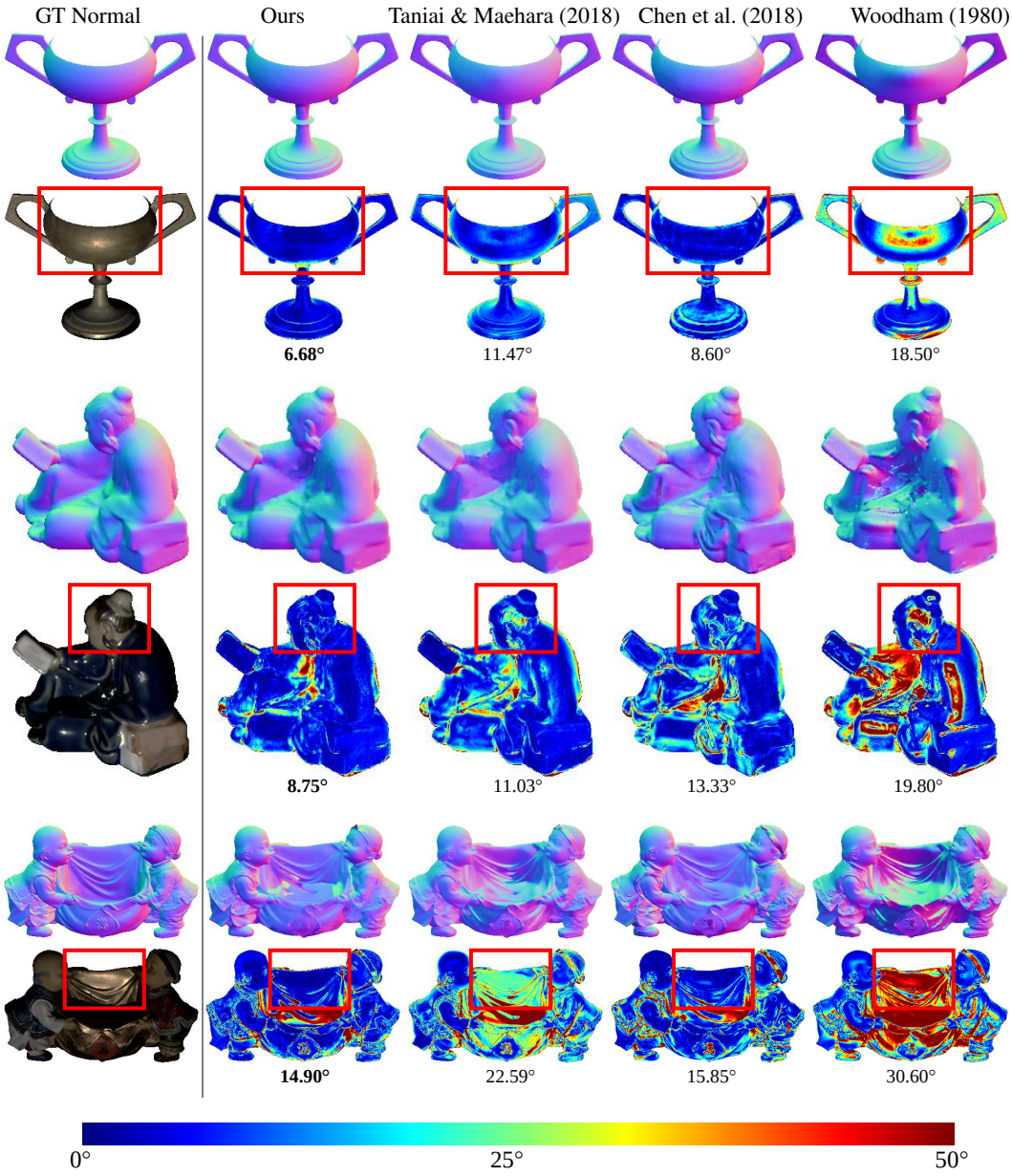


Figure 8: **Normal estimation on specular objects: "Goblet", "Reading" and "Harvest".** As shown in the observed image of these three objects, the "Goblet" is mostly made of metallic materials; "Reading" and "Harvest" present many specular effects over the clothes. Our method achieves the best performance in all these three objects, especially in those regions with high specularities. Please look at the red windows in the error map. "Reading" contains many specularities over its cloth and its head. While all the other methods suffer on these specularities, our method still performs well in these regions, especially on the head. The cloth of "Harvest" in the center also presents significant specular effects. While the other self-supervised method TM18 (Taniai & Maehara, 2018) failed on these regions, our method correctly recover the surface normal. These results demonstrate that our method is taking advantage of the information that the specularities provide. Hence, we can estimate the normal accurately on specular regions.

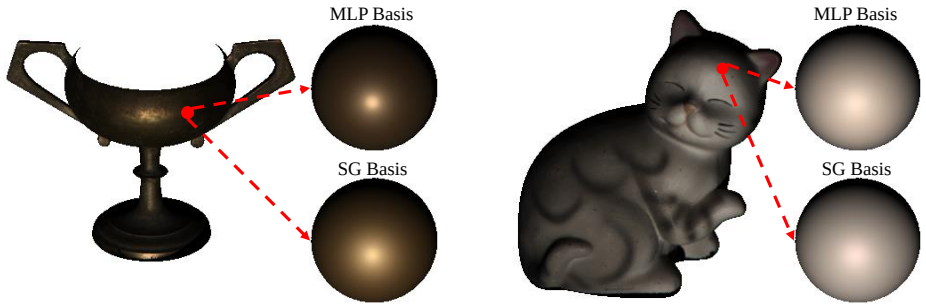


Figure 9: **Comparison on different choices on the specular model.** The above shows the estimated material of our model with different choices on the specular basis. We can see that, from the object "Goblet" on the left side of the figure, the MLP basis is advantageous in representing metallic effect with high-peak and long-tail effects than the SG basis. For objects with soft and broad specular effects, e.g. the ceramic cat on the right of the figure, both models achieve comparable results.

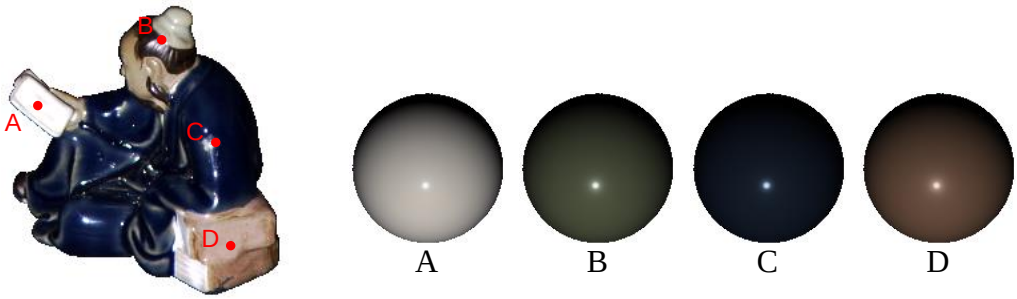


Figure 10: **Visualization on the estimated svBRDFs.** This figure shows the estimated svBRDFs of the surface points on the objects. On the left is the observed image of "Reading". We select four different surface points on the object and showcase our estimated BRDF spheres of those points on the right side. The results demonstrate that our learning model on the BRDF can accurately recover the materials from metallic to diffuse. Noted that, for better visualization, we normalize the BRDF spheres to have the maximum intensity to be 1. The observed images are also scaled up for visualization.

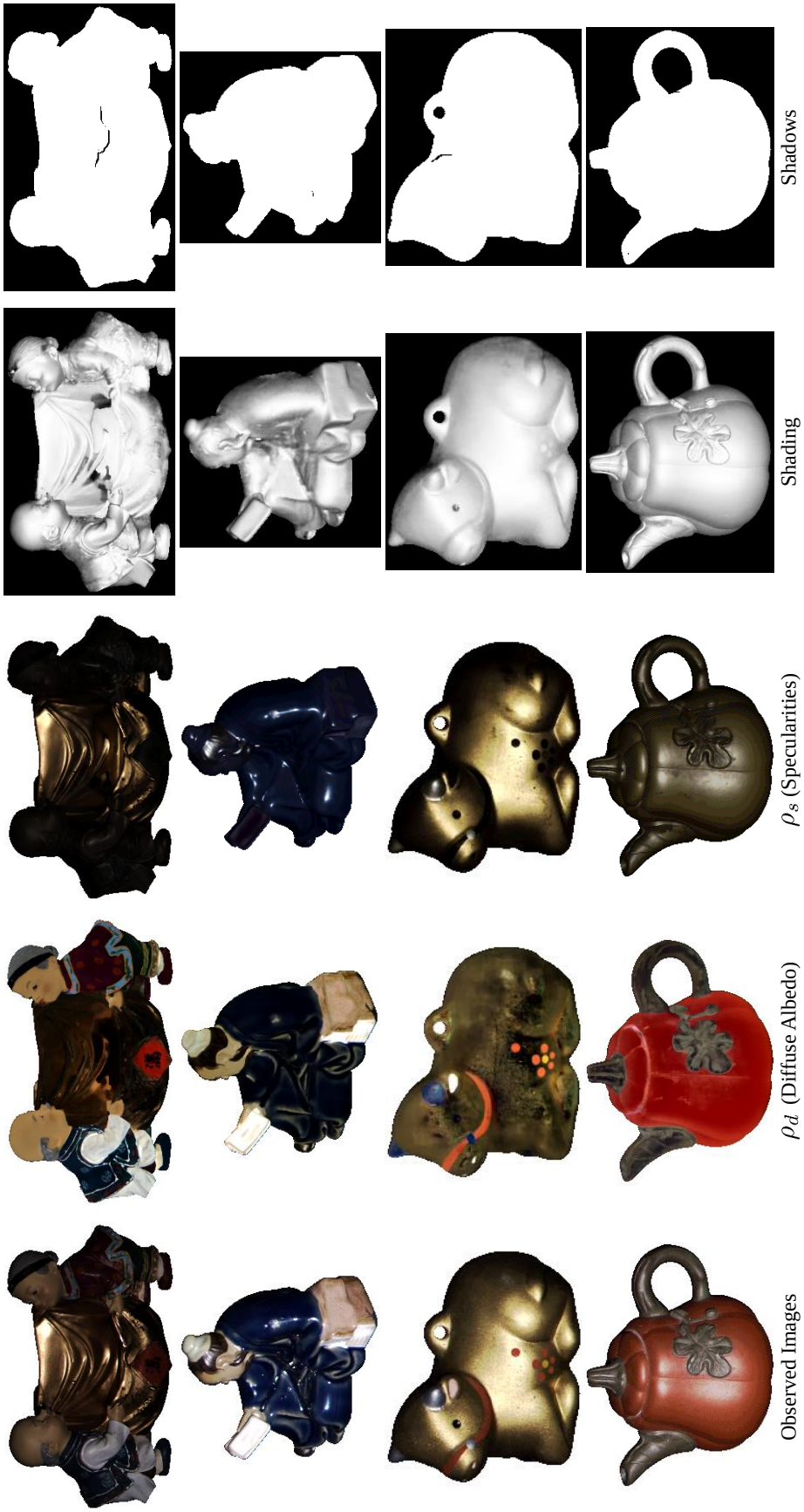
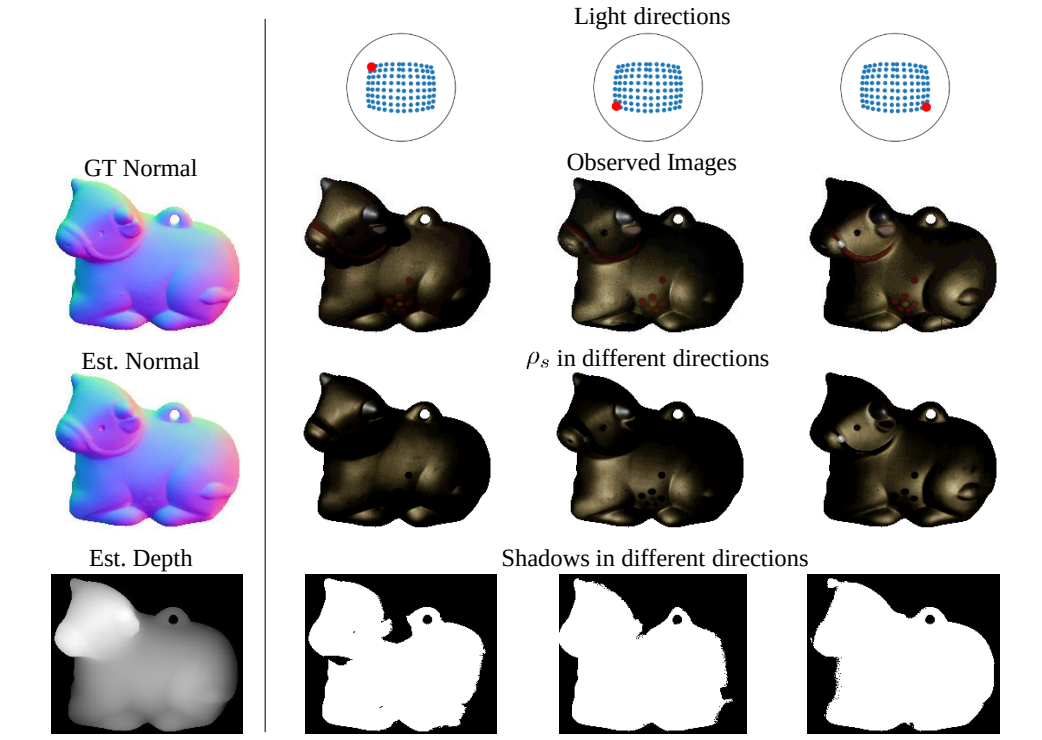
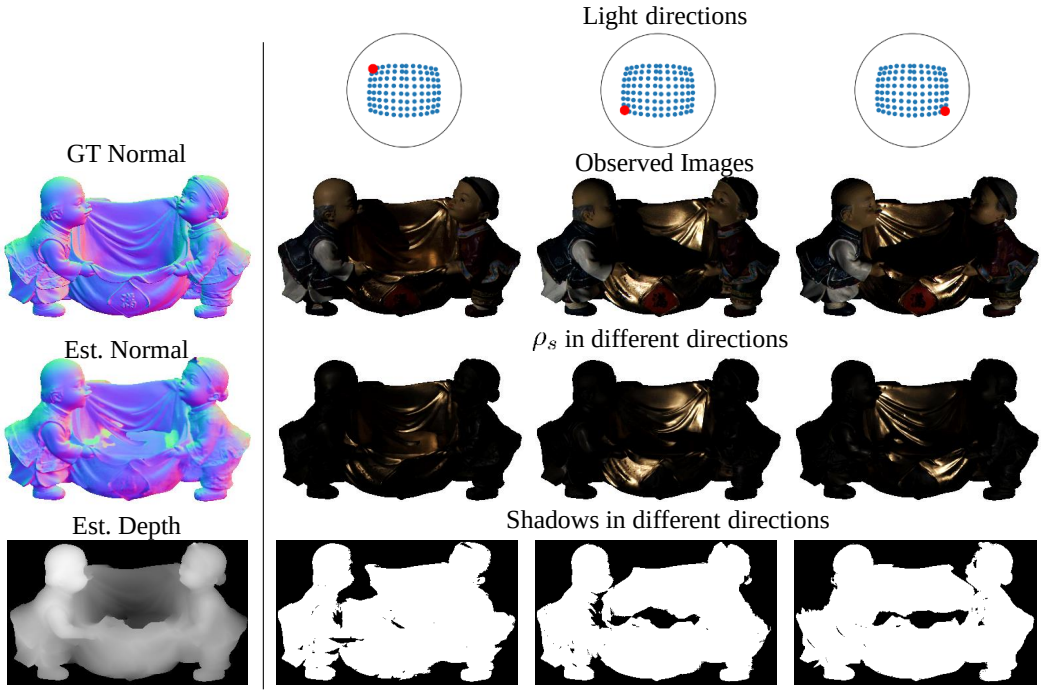


Figure 11: **Visualization on each terms of the rendering equation Eq. (1)**. In the above images, the first column displays the observed images of the objects. The second and third column are the estimated diffuse albedo ρ_d, and specular components ρ_s. The fourth column is the shading map, which is computed by the dot product between light direction and surface normal ($l^T n$). The last column is the estimated shadows, corresponding to s in the equation. As seen from the diffuse albedo map in the cloth of the "Harvest" and the small patterns on the "Cow", the estimated diffuse albedo map retains the objects' fine details. These results demonstrate that our method can recover the fine details of the svBRDF map in the object. Note that, for better visualization, the images we selected here are all illuminated by a front light source. Hence, as can be seen in the observed images, there is little shadows.



(a) Cow



(b) Harvest

Figure 12: **Estimated specularities and shadows under different illuminations.** The leftmost column presents the ground truth normal and our estimated normal and depth as a reference of the object's geometry. We show the estimated specular components ρ_s and estimated shadows under three different extreme lighting directions in the right-three columns. In "Cow", the object is generally smooth, and our estimation of the shadows also visually match the observed images. "Harvest" has a complex geometry and consists of many depth discontinuities over the surface. As discussed in the Sec. 6 of the main paper, our method is influenced by these regions and will generate a "shallower" depth map. Hence, the estimated shadows are generally under-estimated.

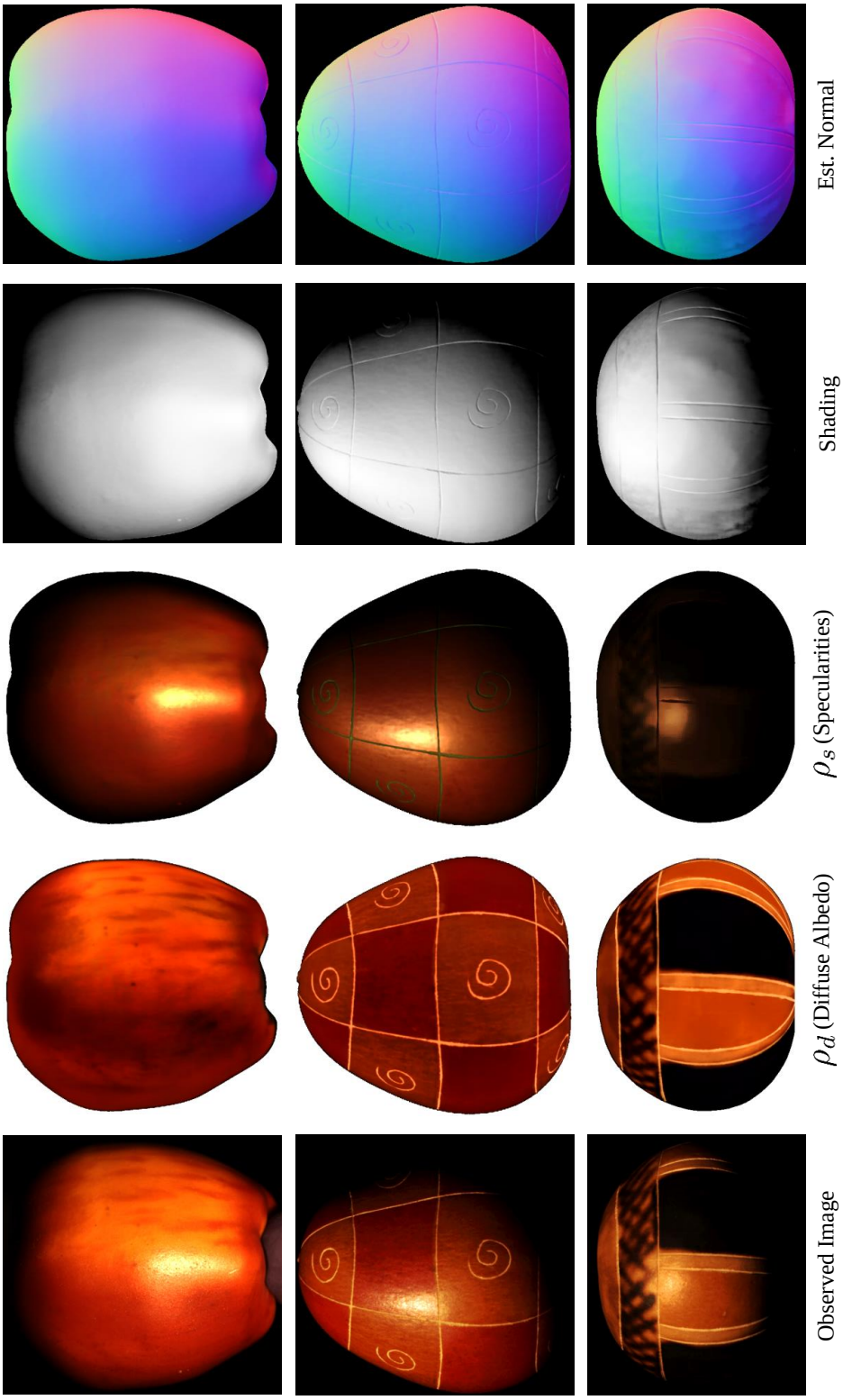


Figure 13: **Results on Gourd&Apple dataset** (Alldrin et al., 2008). The columns from left to right are the observed images, our estimated diffuse albedo, specularities, shading, and surface normal of the objects.

Figure 14: **Results on Light Stage Data Gallery** (Chabert et al., 2006). The columns from left to right are the observed images, our estimated normal, diffuse albedo, and specularities of the objects.