# A Neural Height-Map Approach for the Binocular Photometric Stereo Problem

Fotios Logothetis
Cambridge Research Laboratory,Toshiba Europe Ltd.
Cambridge, UK
fotios.logothetis@toshiba.eu

Ignas Budvytis
University of Cambridge
Cambridge, UK
ib255@cam.ac.uk

Roberto Cipolla
University of Cambridge
Cambridge, UK
rc10001@cam.ac.uk

## Abstract

*In this work we propose a novel, highly practical, binocular photometric stereo (PS) framework, which has same acquisition speed as single view PS, however significantly improves the quality of the estimated geometry.*

*As in recent neural multi-view shape estimation frameworks such as NeRF [29], SIREN [35] and inverse graphics approaches to multi-view photometric stereo (e.g. PS-NeRF [38]) we formulate shape estimation task as learning of a differentiable surface and texture representation by minimising surface normal discrepancy for normals estimated from multiple varying light images for two views as well as discrepancy between rendered surface intensity and observed images. Our method differs from typical multi-view shape estimation approaches in two key ways. First, our surface is represented not as a volume but as a neural heightmap where heights of points on a surface are computed by a deep neural network. Second, instead of predicting an average intensity as PS-NeRF or introducing lambertian material assumptions as Guo et al. [7], we use a learnt BRDF and perform near-field per point intensity rendering.*

*Our method achieves the state-of-the-art performance on the DiLiGenT-MV dataset adapted to binocular stereo setup as well as a new binocular photometric stereo dataset - LUCES-ST.*

## 1. Introduction

Single view Photometric Stereo is a long standing problem in Computer Vision. Recent methods [7, 8, 21] have achieved impressive normal estimation accuracy on both real and synthetic [9, 21] datasets. However, the progress in the quality and practical usefulness of the estimated shape (e.g., by using the numerical integration of [32]) has been much less convincing, due to the heavily ill-posed nature of the global shape estimation problem (see Figure 1).

One way to improve the quality of the global shape extracted from Photometric Stereo images is to leverage multiple views. A classical approach to multi-view photometric stereo [19] involves obtaining initial sparse point-cloud via structure from motion [34]. Depths of these points are then propagated along the iso-depth contours in each view. Not having any learnable component this approach is fragile to errors in computation of the iso-depth contours. Recently proposed PS-NeRF [38] reformulated the multi-view photometric stereo problem as an inverse graphics problem by learning a neural textured volume to minimise discrepancy between estimated surface normals and predicted photometric stereo normals as well as predicted intensities and observed intensities. It unsurprisingly achieves the state-of-the-art on the DiLiGenT-MV [19] multi-view photometric stereo benchmark.

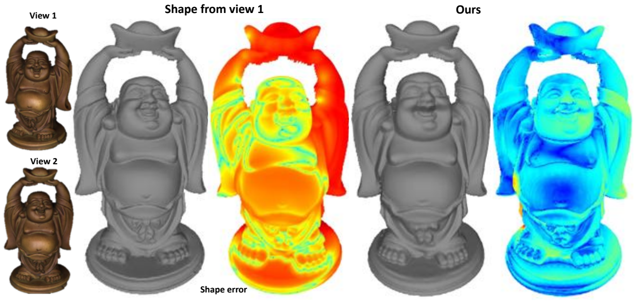

Figure 1. This figure illustrates the limitations of the use of the single view photometric stereo for shape estimation. The left pair of images show a plausible shape estimated using single view PS method of [21] which respects estimated normals well (zoom in for a better view), however its actual shape error is large due to bending (right side of the statue) caused by shape discontinuities or small systematic errors in estimated normals. In contrast, our Binocular Photometric Stereo method while having the same capture time and little hardware costs obtains a significantly improved shape. Error maps are visualised by using jet color scheme on per-pixel shape error. All pixels with an error above 1.5mm are assigned with dark red color.

While multi-view photometric stereo can achieve low reconstruction errors (few tenths of milimeters), for some applications such as robotic interaction, and conveyor belt scanning, it is not feasible due to long capture times and precise camera pose calibration (especially when a single camera is used) required. Hence in this work we consider the binocular photometric stereo setup where multiply lit images are obtained for a pair of cameras. Note that Binocular Photometric Stereo has been introduced in [15] and subsequently developed in [3, 37] under different assumptions and for different applications. However the aforementioned methods do not model materials with complex reflectances, e.g., highly specular materials such as metal or porcelain.

To address this limitation, we adapt the recently popular neural rendering approaches (e.g. NeRF [29], SIREN [35], PS-NeRF [38]) to the binocular photometric stereo setup. Note while it is a popular belief that NeRF-like approaches do not work well in sparse setup we show that two views are enough to compute accurate shape (see Tables 1 and 2) if care is taken in modelling of the neural representation of shape and losses used. In particular, instead of a neural density [29] or signed distance field [35] we leverage a neural heightmap where heights of points on a surface are computed by a deep neural network. In comparison to volume-based shape representations, this allows for better conditioned and efficient surface optimisation procedure. Moreover, instead of predicting an average intensity as PS-NeRF[1] or introducing lambertian material assumptions as Guo et al. [7], we use a learnt BRDF and perform near-field per point intensity rendering.

In more detail, our method works by combining three steps of: (1) estimating per-view based shape by using per view estimated photometric stereo normals [21] and (2) using it to initialize neural heightmap network guided by estimated pixel-wise normals and depth (initialising the albedo value to a constant) and (3) fitting the initialised neural heightmap to image intensity and estimated normal maps.

Our method achieves the state-of-the-art performance on the DiLiGenT-MV [19] dataset adapted to binocular stereo setup. It is also evaluated on a new binocular photometric stereo dataset, LUCES-Stereo, consisting of 7 objects from original LUCES [26] dataset captured in the binocular photometric stereo setup. See Sections 4.1 and 5 for more details. Our contributions include:

- A neural height-map approach to the Binocular Photometric Stereo problem which is robust to highly complex materials
- A Binocular Photometric Stereo dataset - LUCES-ST.

## 2. Related Work

There is an extensive literature leveraging photometric cues for single and multi-view based 3D reconstruction. Here we categorise as follows.

**Single view photometric stereo.** Recently deep PS has been very successful on solving the single view far-field PS problem from CNN-PS [9] to PX-Net [20] which is also extended for the near-field setting [21]. Other works like [10] incorporated material reflectance priors for single view normal prediction or used specific BRDFs [5, 6, 22, 24, 27], including Lambertian or Ward reflection models. Other recent approaches have also tackled a more uncalibrated setting like [2, 17, 18, 39]. Finally, [7] introduced the idea of a infinitely differentiable surface (SIREN [35]) with Lambertian rendering to directly optimise a neural surface from intensities. Our method is similar to [7] but extended in a stereo setting and with a non-Lambertian rendering.

**Sensor enhanced photometric stereo.** Some works have also utilised various 3D scanning techniques such as laser scanner and structured light [16, 33, 40] allowing to fit reflectance functions at each surface point. While it may be possible to combine photometric stereo with structured light scans [1, 30], accurately merging RAW data from different type of scans is a challenging task and can limit the resolution of the reconstruction

**Binocular photometric stereo.** Specific binocular Photometric Stereo has been introduced in [15] and subsequently developed in [3, 37] under different assumptions and for different applications. The limitations in these cases are the lack of generality in terms of material reflectance which makes these methods not being very effective with specular outliers.

**Multi-view photometric stereo.** [23] proposed a multi-view Photometric Stereo which retrieved the volume with the sign distance function based parameterisation [31, 41]. Such approach relied on structure-from-motion initialisation and the photometric refinement used diffuse image irradiance equations.

Similarly, [19] introduced a method for capturing both 3D shape and reflectance with a multi-view photometric stereo setup. The idea is to collect photometric stereo images multiple viewpoints and combine it with structure-from-motion to obtain a precise reconstruction of the complete 3D shape. The spatially varying isotropic bidirectional reflectance distribution function (BRDF) is captured by simultaneously inferring a set of basis BRDFs and their mixing weights at each surface point.

Recently, neural surface approaches have become very popular from the introduction of NeRF [29]. This has been extended to neural SDF approaches like [11–13] and very recently to more structured rendering approaches like Ref-NeRF [36] and PS-NeRF [38].

[1] Note, PS-NeRF [38] runs in two stages. First stage uses average images and renders intensities per light image only in the second stage during which the shape is not updated.

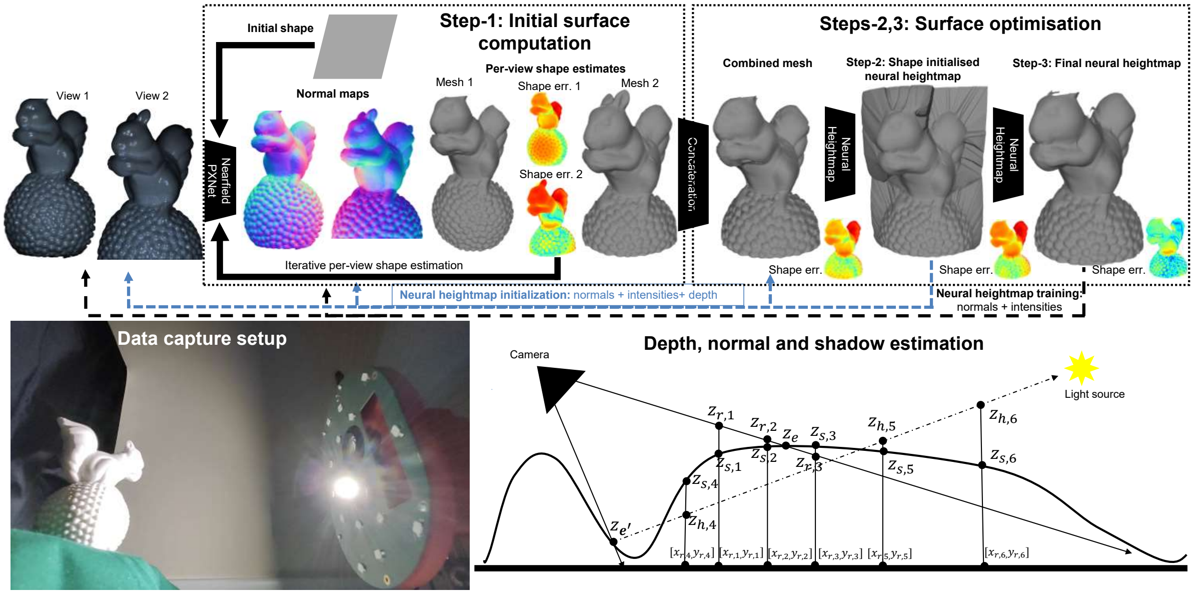

Figure 2. Graphic illustration of the proposed approach. The bottom left shows our binocular photometric stereo data capture setup. The top figure illustrates three key steps of our method: (1) joint normal and shape estimation for each view using [21], (2) initialisation the neural heightmap based on SIREN [35] architecture using the shape and normals estimated in step 1 and (3) the main training step of the neural heightmap. The bottom right part of this figure illustrates the ray sampling procedure used to compute normal, depth and shadow estimates from the heightmap. Note that sample points 1-3 and 4-6 correspond to 2 different surface points $z_e$ and $z_{e'}$.

## 3. Method

Our binocular photometric method consists of three key steps. First step involves joint normal and shape estimation for each view independently and is described in Section 3.1. Second step uses the estimated shape to initialise a neural heightmap described in Section 3.2. Finally, the initialised neural heightmap is trained, using losses described in Section 3.3 to explain observed photometric stereo image intensities and estimated normals as explained in Section 3.4.

### 3.1. Per-view shape estimation

We start by computing per view normal maps using the state-of-the-art PS normal estimation network - PX-Net [21]. This method offers a general near-field network that obtains high quality normal maps for the calibrated, near (and far) field PS setting as well as some reasonable surface estimate. Qualitative examples of the shape obtained using [21] for camera 1 are shown in Figure 4 (see column *Logothetis et al.*). We use the estimated per view shape to initialise neural heightmap as described in Section 3.4. Note, as shown in Figures 2 and 4, high quality local shape is obtained whilst suffering from global bending due to the ill-posed nature of shape estimation under discontinuities or systematic error in estimated normals. Note the step of per view shape from normal estimation is crucial to speeding up the recovery and constraining of the neural surface as purely relying on PS image intensities from sparse view-points is likely to take a significant training time and obtain suboptimal surface as indicated by some results discussed in Section 5.

### 3.2. Neural heightmap

**Surface parameterisation.** We start by assuming that the surface can be expressed as a continuous height map $z_s = F(x_s, y_s)$ in some word coordinate system (we use the subscript $s$ to denote surface coordinates). For the case of stereo cameras, this coordinate system is chosen as the average between the 2 camera system (i.e. the 'rectified' stereo system). We note that a roto-translation ($R_c$, $\mathbf{t}_c$) is required to convert between this coordinate system and the original camera coordinate system i.e.:

$$[x_s, y_s, z_s]^\intercal = R_c \cdot [x_c, y_c, z_c]^\intercal + \mathbf{t}_c \tag{1}$$

The unknown function $F$ is a deep neural network and the objective is to optimise its weights. Extending [7] to the 2 view problem, we chose the SIREN architecture [35] which is an MLP with sinusoidal activation functions and that guarantees that the surface is infinitely differentiable thus can be easily recovered from its derivatives; thus the surface normal is $\mathbf{n}_s \propto [\frac{\partial F}{\partial x_s}, \frac{\partial F}{\partial y_s}, -1]^\intercal$ and automatic differentiation makes it a function of the network weights. We also add a scalar (grayscale) albedo $\rho = F(x_s, y_s)$ channel on the SIREN used for rendering.

**Surface sampling**. We note that since the projection depth

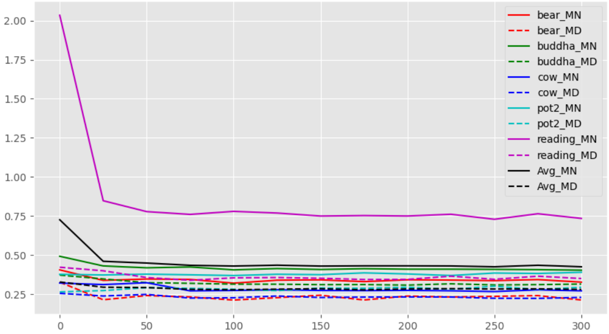

Figure 3. This figure plots the mean and median shape error for all DiLiGenT-MV objects. It is observed that we achieve quick convergence with minimal improvement after a few tens of epochs, where each epoch takes roughly 1 minute on an unoptimised code run on a single RTX4080 GPU.

$z_c$ in Equation 1 is unknown, exact conversion between coordinate systems is impossible. This is a clear difference to the single view of [7] where one-to-one mapping between image and depth exists (this can never be the case in stereo due to left-right occlusions). To overcome this issue, a number of tentative depth samples $z_{c1}, z_{c2}, z_{c3}, \dots$ are considered and are used to generate points $i$ along the viewing direction $\mathbf{v}$ as $\{\mathbf{v}z_{ci}\}$. Applying the coordinate transfer Equation 1 gives the coordinates of these points in the world space as $\{[x_{ri}, y_{ri}, z_{ri}]\}$ . This is visualised in Figure 2. Then, the network function $F$ can be queried in the position $\{[x_{ri}, y_{ri}]\}$ to get the surface depth estimates $\{z_{si} = F[x_{ri}, y_{ri}]\}$. Finally, in order to get the 'actual' depth estimate $z_e$, the set of depth estimates is reduced with volumetric rendering, using as opacity $\alpha_i$ the inverse of the depth squared difference between $z_{si}$ and $z_{ri}$, i.e. $\alpha_i = \exp(-f(z_{si} - z_{ri})^2)$ , with $f$ being a scaling factor used to convert between millimeters and normalised units. The volume rendering equation then becomes: $z_e = \sum_i \left(z_{si}\alpha_i \sum_i (1 - \alpha_{i-1})\right)$. We note that the surface normal $\mathbf{n}_e$ and albedo $\rho_e$ are computed with a similar volume rendering equation using the same opacity $\alpha_i$.

**Intensity rendering.** To render light intensities, we first need to compute the near-field lighting vectors $\mathbf{l_m}$ and light attenuation $a_m$ (for light source $m$). Following the near lighting model from [28], for calibrated point light sources at positions $\mathbf{s}_m$, each surface point $\mathbf{p}$ gets variable lighting vectors $\mathbf{l}_m = \mathbf{s}_m - \mathbf{p}$ and attenuation factors $a_m(\mathbf{p}) = \phi_m \frac{(\hat{\mathbf{L}}_m(\mathbf{X}) \cdot \hat{\mathbf{d}}_m)^{\mu_m}}{||\mathbf{l}_m(\mathbf{p})||^2}$ where $\hat{\mathbf{l}}_m = \frac{\mathbf{l}_m}{||\mathbf{l}_m||}$ is the lighting direction, $\phi_m$ is the intrinsic brightness of the light source, $\hat{\mathbf{d}}_m$ is the principal orientation of the LED and $\mu_m$ is an angular dissipation factor.

Then, the total intensity $i_m$ is computed as $i_m = s_m \cdot a_m \cdot \rho \cdot \mathbf{BRDF}(\mathbf{n}, \mathbf{l_m}, \mathbf{v})$. Here $s_m$ is a 'soft' indicator variable that is 0 for shaded points and 1 otherwise (see bellow).

**Learned BRDF renderer.** Our aim is to learn a single BRDF model (assuming uniform material with potentially varying albedo) following the principles described in the MERL real material database [25]. For that, the half vector $\mathbf{h} = \frac{\mathbf{l_m}+\mathbf{v}}{|\mathbf{l_m}+\mathbf{v}|}$ is first computed as well as the relatives angles between $\mathbf{n}, \mathbf{h}$ and $\mathbf{l}$ namely $\theta_h, \phi_h, \theta_d, \phi_d$ (see supplementary for more details). Moreover, it is desired to only recover isotropic materials there $\phi_h$ is ignored. In addition, real BRDFs follow the Helmholtz reciprocity constraint and so $\text{BRDF}(..., \phi_d) = \text{BRDF}(..., \phi_d + \pi)$.

Thus, we parameterised

$$\text{BRDF}(\mathbf{n}, \mathbf{l_m}, \mathbf{v}) := (\mathbf{n} \cdot \mathbf{l_m})\text{MLP}(\theta_h, \theta d, \phi_d) \quad (2)$$

we use 3x16 hidden layers with relu activation and exponential activation (the BRDF values must be always nonzero and should be around 1 for diffuse materials) for the output layer. We note that even though the surface point is computed as a weighted sum over the ray (as above), the renderer is only computed on a single sample.

**Shadow estimation.** To estimate cast shadows, we raytrace from each surface point to the light source following the direction of the lighting vectors $\mathbf{l_m}$ computed above. For each ray we take 16 samples $h$ every 1.5mm starting 3mm away from the start. For all these points, we query the depth of the height map and compute the difference $d_h = z_h - z_s$; if at least one of these differences is negative, there is a shadow. This shadow computation can be differentiably approximated as: $\text{SM}\Big(-\text{sigmoid}(d_h)\Big)$ (where SM denotes the *softmax* operator).

### 3.3. Losses

Our neural surface is trained with the following losses.

**Angular normal loss.** We apply normal loss on surface normals to match single view normal estimates (from [20]) using the angular loss formula:

$$L_n = |\text{atan2}(||\mathbf{n}_n \times \mathbf{n}_s||, \mathbf{n}_n \cdot \mathbf{n}_s)|\max(\mathbf{n}_n \cdot \mathbf{v}, 0) \quad (3)$$

For experiments where this loss is used, the relative weighting of this loss is 1 (with normals measured in degrees).

**Rendering loss.** We include an L1 error on the rendered intensities (for all lights $m$) as: $L_r = ||i_{t,m} - i_{r,m}||$. Relative weights are 100 for LUCES-ST and 1000 for DiLiGenT-MV (image values are rendered in [0,1] and which has DiLiGenT-MV darker images).

**Depth loss.** Used only at the initialisation stage (as depth estimates are very inaccurate) $z_s$, $L_z = \lambda_z|z_s - z_t|$. Relative weight is 1 (with depth in mm).

**Regulariser.** For numerical stability reasons, we apply normal and depth regularisers ($\mathbf{n} = [0, 0, 1]$, $z = \text{mean}(z_0)$) with respective weights 1e-3 and 1e-4.

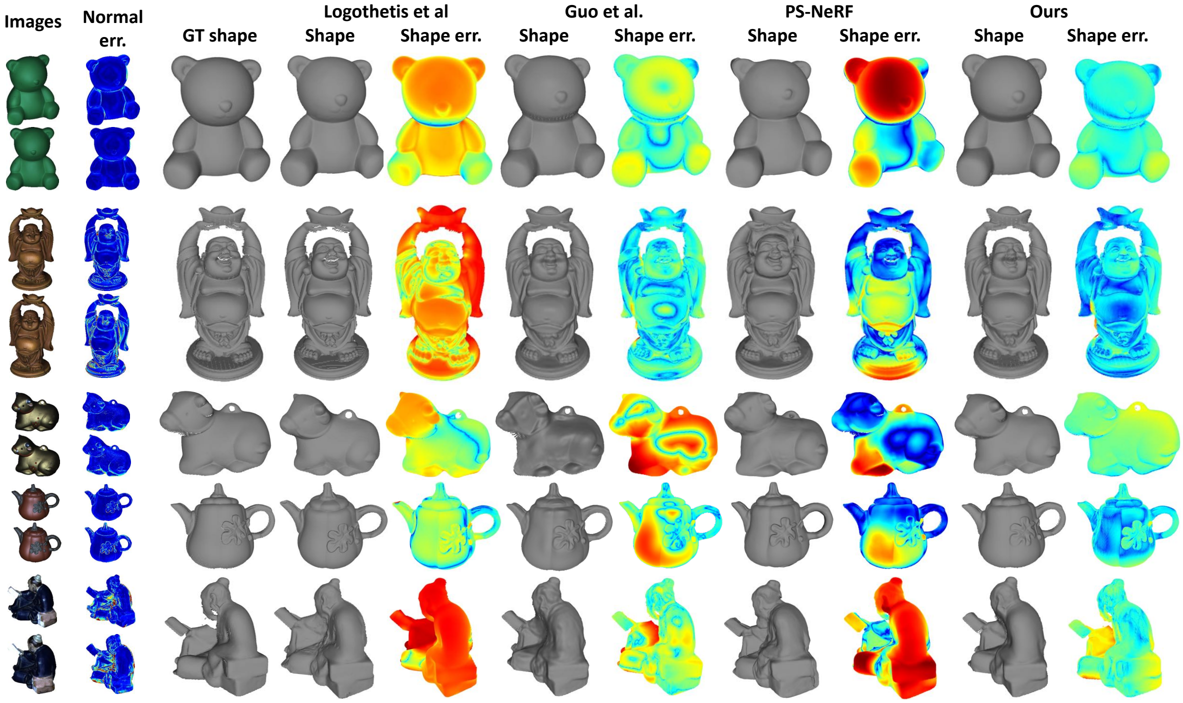

Figure 4. This figure shows qualitative results on DiLiGenT-MV [19] objects for our Logothetis et al [21] (single view - camera 1), adaptation of Guo et al. [7] to two view shape estimation, PS-NeRF [38] and ours. For each of the method we visualise the predicted shape and predicted shape error map using jet color scheme (errors above 1.5mm have saturated red color). Our method outperforms all methods. Also note, that as expected, the adaptation of Guo et al. [7] fails on cow object significantly as it is a highly specular object. PS-NeRF seems to have its reconstructions divided into strongly correct (top of *Pot2*) and strongly incorrect (bottom of *Pot2*) regions which is likely due to the use of systematically incorrect normals.

**Sample weighting.** To minimise the impact of self-reflections, we note that these tend to occur on oblique points where there are also shadows. Thus, for each point the *approximate ambient occlusion* $a$ (measure of obliqueness, see [4]) is computed as $a = \frac{\text{number of shadows}}{\text{number of lights}}$. We multiply normal loss with $a$ and rendering loss by $a^2$ as intensities are less robust to self-reflections than normal estimates

**Implementation details.** We use official tensorflow implementation of SIREN with $5 \times 512$ layers and 1.05M parameters. For the first layer, we use a 50 frequency in DiLiGenT-MV and 100 in LUCES-ST (due to higher resolution input images).

### 3.4. Training

**Initialisation stage.** We note that surface sampling procedure described above requires that the surface is appropriately initialised for shadow computations to be meaningful. To achieve this, we project the initial surface (obtained from [21]) into the unified coordinate system and then pre-train just the SIREN function with normal and depth loss. Of course the initial depth maps are inconsistent and the network at best can converge to an estimate of their average. We also apply data augmentation of $\pm 1mm$ on word coordinate points at that stage in order for the initial surface to be smooth (so that the network does not attempt to make two copies of the surface). We train the pre-initialisation stage for 300 epoch on DiLiGenT-MV and 30 on LUCES-ST which takes around 5 mins on GTX4080 (batchsize 16384). Note that no-ray sampling, rendering and shadows is used in the stage therefore allowing for much higher batchsize and much faster epochs than the main stage.

**Training stage.** During the main training stage, we use 128 depth samples per ray and 16 shadow samples. We train with batchsize 512 and 1024 on DiLiGenT-MV and LUCES-ST respectively (DiLiGenT-MV has 96 vs 15 lights increasing memory consumption). We run for 300 epochs on DiLiGenT-MV and 50 epochs on LUCES-ST (LUCES-ST objects containing around 1.5M samples with contrast to around 100K for DiLiGenT-MV). To further increase the convergence speed, we only enable rendering and shadows after epoch 2 on LUCES-ST and 10 on DiLiGenT-MV (so the first epochs run with normal loss only and take around half the time). Also see Figure 3.

| Method | Bear | Buddha | Cow | Pot2 | Reading | Average SE | Median SE |
|---|---|---|---|---|---|---|---|
| DiLiGenT-MV [19] [all views] | 0.74 | 0.53 | 0.83 | 0.57 | 1.39 | 0.81 | 0.23 |
| PS-NeRF [38] [all views] | 0.45 | 0.40 | 0.58 | 0.40 | 0.61 | 0.49 | 0.31 |
| Logothetis et al. [21] [1 view - camera 1 only] | 2.62 | 2.77 | 1.14 | 0.89 | 6.32 | 2.75 | 2.41 |
| Logothetis et al. [21] [2 views] | 2.70 | 3.23 | 0.87 | 0.79 | 5.97 | 2.71 | 2.51 |
| PS-NeRF [38] [2 views] | 2.64 | 1.02 | 1.02 | 0.94 | 3.88 | 1.90 | 1.57 |
| Guo et al. [7]* [2 views] | 0.86 | 0.51 | 3.21 | 1.39 | 1.05 | 1.40 | 1.18 |
| Ours - [PS-NeRF [38] normals only] | 1.17 | 0.57 | 0.82 | 0.78 | 0.90 | 0.85 | 0.63 |
| Ours - [normals only] | 0.40 | 0.49 | 0.60 | 0.34 | 0.75 | 0.52 | 0.38 |
| Ours - [intensities only] | 0.73 | 0.57 | 0.69 | 0.61 | 0.87 | 0.69 | 0.58 |
| Ours - [PS-NeRF [38] normals + intensities] | 0.66 | 0.51 | 0.65 | 0.68 | 0.81 | 0.66 | 0.51 |
| Ours - [normals + intensities ] | 0.57 | 0.51 | 0.75 | 0.56 | 0.75 | 0.63 | 0.50 |

Table 1. This table shows both ablation (last 5 rows) and main results on DiLiGenT-MV [19] dataset. For all objects we report the mean shape error as well as average shape error and average median shape error on all objects. All of our experiments are performed using first 2 views but single view and all view competitors are shown for reference. In the ablation experiment we run our method using single view normal map from PX-Net [21] (current calibrated PS SOTA) or [38] (originally computed with [2] using all 20 views) in order to have a fair comparison with [38]. In addition, the effectiveness of intensity rendering is also ablated for both input normal configurations as well on its own. We note that our best configuration on this experiment is using normals only and significantly outperforms all other 2 view competitors (0.52mm average SE vs 1.4mm for [7]) and its only marginally worse than the 20view SOTA (0.49mm for [38]). We emphasise that all of our ablation experiments are also significantly outperforming all other competitors showing the strength of our approach. Finally, it is interesting to note that the intensity rendering is only improving performance when combined with [38] normals and it is actually decreasing performance compared to SOTA (PX-Net) normals only. This is probably due to inaccurate near-lighting modeling on DiLiGenT-MV [19] as no light angular dissipation factors $\mu$ are provided. In contrast, on the truly near-field LUCES-ST 2, intensity rendering improves in most cases.

## 4. Experimental Setup

In this section we describe the datasets used in our evaluation, including the new LUCES-ST dataset. We also discuss evaluation metrics and various implementation details.

### 4.1. LUCES-ST dataset

We present LUCES-ST dataset with a subset of 7 objects: *Bell*, *Bunny*, *Cup*, *Hippo*, *Owl*, *Queen*, *Squirrel*, out of the original 14 of [21]. We note that we only reused the objects and their CT scanned GT meshes and all of the stereo capture data is new. We use a stereo capture device with 2x - Flea3 FL3-U3-32S2C-CS 1/2.8" Color USB 3.0 Camera pointgray cameras (2080 × 1552 px) and 15 LED lights as shown in Figure 2. We use 8mm lenses for the cameras and place the objects around 15-20 cm away from the camera in order for the near lighting effects to be significant (as opposed to DiLiGenT-MV [19]). This sparse lighting setting makes the photometric stereo problem extra challenging adding to the value of our dataset. Note that as the 2 stereo images per light are captured simultaneously, the effective light directions for each pixel differ for the 2 views (due to parallax) in contrast to turntable setups like DiLiGenT-MV. This makes application of uncalibrated PS methods (such as [2]) that rely on the lighting vectors being the same at each view) extra challenging. The data is available to download at https://www.toshiba.eu/pages/eu/Cambridge-Research-Laboratory/luces.

### 4.2. Metrics and comparison details

**Performance metrics.** As we only use a stereo pair of views to compute reconstructions, full object Hausdorff distances are not informative. Therefore, in order to have a fair comparison, we compute a cropped ground truth (though back-projection of the ground truth depth maps) and the compute Hausdorff distance *from the ground truth* to the reconstructed objects. That makes sure that the metric is fair for all competitors producing variable size outputs. Therefore, computed error maps are all shown on the GT and hence are comparable between different competitors.

**Datasets.** Along with LUCES-ST, we evaluate our approach on the synthetic version of LUCES-ST (see Table 2 and supplementary material) as well as a popular real multi-view PS benchmark - DiLiGenT-MV [19]. DiLiGeNT contains 5 objects *Bear*, *Buddha*, *Cow*, *Pot2* and *Reading* captured from 20 views, with 96 light images of 612 × 512 resolution. The objects are around 1.5 m away from all cameras (turntable capture setup) and the camera focal length of 50 mm approximates orthographic viewing. As we are only interested in a single pair of views, we chose the first 2 views.

**Adapting competing methods to binocular PS setup.** No recent method has focused on the Binocular PS problem so fair comparison is non trivial. We compare with [21] which shows SOTA performance on single view and to compute a 2 view result, the 2 independent view reconstructions are

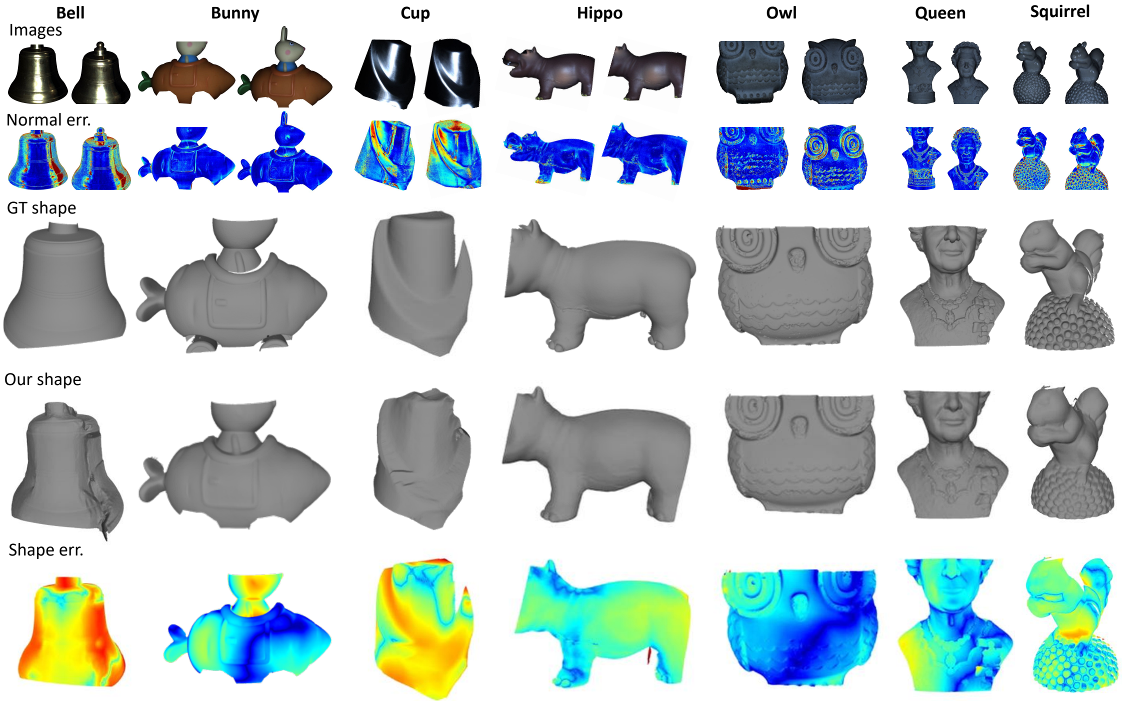

Figure 5. This figure shows the qualitative results of our method on LUCES-ST dataset. The first thre rows show the cropped images and corresponding error images of normals estimated from PX-Net [21] and the ground truth shape. The final two rows show the shape predicted by our method and corresponding error map (from ground truth to reconstruction). As in Figure 4 any errors larger than 1.5mm are clamped to a dark red color. Our method performs well on *Hippo*, *Owl*, *Queen* and *Squirrel* objects as also reflected in Table 2. Performance is worse on highly specular, metallic objects such as *Bell* and *Cup*.

concatenated and merged with Poisson reconstruction [14]. The very recent single view method of [7] is conceptually our closest match due to the use of the SIREN surface and rendering. To simulate their 2 view extension, we adapt our implementation to perform pure Lambertian rendering and disabled the normal loss and shadow computation. We note that the same initialisation with normals and average surface that we used is also used for them. Finally, we compare to PS-NeRF [38] which has available code online. We note that the surface is not updated during their second stage of training (which optimises re-rendering) so for the purpose of surface error, stage 1 is only used. That means that normal maps and average intensity images are only used. Therefore, to apply them on LUCES-ST, the normal maps of [21] are used.

## 5. Experiments

In this section we report results on DiLiGenT-MV [19] and LUCES-ST datasets. This is shown quantitatively on Tables 1 and 2 and visualised in Figures 4 and 5 respectively. Also the use of intensity rendering is ablated for both datasets by provided results with normal only, intensity only and combined losses.

**DiLiGenT-MV [19] experiments.** We note that our best configuration on DiLiGenT-MV [19] (see Table 1) turns out to be using normal loss only and significantly outperforms all other 2 view competitors (0.52mm vs 1.4mm average SE for [7]) and is only marginally worse than [38] using 20 views (0.49mm). In addition, we note that all of our configuration experiments with both sets of normal map inputs (ie. normals used by PS-NeRF [38] vs PX-Net [21]) and with/without intensity also outperform all other competitors and achieve less that 1mm in almost all experiments. It is also notable that the use of intensity rendering seems to degrade the performance when combined with PX-Net normals but offers a small improvement when combined with the less accurate [38] normal input. In contrast, in the LUCES-ST experiments (see bellow) intensity rendering offers a clear advantage.

The degradation of performance using intensity rendering on DiLiGenT-MV [19] can be attributed to potentially inaccurate near-lighting modeling, as no light angular dissipation factors $\mu$ are provided. In fact, DiLiGenT-MV [19] provides point light positions and far-field equivalent light intensities, therefore to apply the near-field lighting model ,$\mu = 0$ was assumed and light intensities were compensated with inverse square of average object distance. The last step can be inaccurate depending on exactly how the far-field equivalent light intensities were measured. In contrast, nor-

| Method | Synthetic / Real | Bell | Bunny | Cup | Hippo | Owl | Queen | Squirrel | Average SE | Median SE |
|---|---|---|---|---|---|---|---|---|---|---|
| Ours - [normals only] | Synthetic | 0.90 | 1.37 | 1.18 | 0.79 | 0.78 | 0.46 | 0.60 | 0.87 | 0.57 |
| Ours - [intensities only] | Synthetic | 0.73 | 0.70 | 1.24 | 0.36 | 0.36 | 0.32 | 0.27 | 0.57 | 0.24 |
| Ours - [normals + intensities ] | Synthetic | 0.75 | 0.82 | 1.20 | 0.76 | 0.70 | 0.44 | 0.55 | 0.75 | 0.51 |
| Logothetis et al. [21] [2 views] | Real | 1.24 | 1.14 | 0.69 | 0.62 | 0.64 | 0.97 | 1.92 | 1.03 | 0.78 |
| PS-NeRF [38] | Real | 1.35 | 0.76 | 1.11 | 1.40 | 0.88 | 0.58 | 0.95 | 1.00 | 0.85 |
| Guo et al. [7]* | Real | 4.53 | 1.10 | 3.41 | 0.71 | 0.38 | 0.61 | 0.85 | 1.66 | 1.83 |
| Ours - [GT normals only] | Real | 0.18 | 0.71 | 0.34 | 0.23 | 0.18 | 0.28 | 0.41 | 0.33 | 0.25 |
| Ours - [normals only] | Real | 1.5 | 0.92 | 1.65 | 1.10 | 0.49 | 0.62 | 0.61 | 0.98 | 0.77 |
| Ours - [intensities only] | Real | 1.87 | 1.11 | 1.33 | 0.50 | 0.35 | 0.63 | 0.55 | 0.91 | 0.80 |
| Ours - [normals + intensities] | Real | 1.41 | 1.01 | 1.74 | 1.05 | 0.46 | 0.62 | 0.59 | 0.98 | 0.77 |

Table 2. This figure shows the quantitative results of our method on LUCES-Stereo dataset. We show comparisons with [21], [38] and [7] (adapted to Binocular Photometric Stereo setup) as well as ablations of the use of intensity rendering. In addition, we also provide an ablation of our method on synthetic version of LUCES-Stereo dataset (containing Blender renderings of the same objects with different pose and segmentation masks, see supplementary Figure 1). Finally, results using ground truth normals as an input are also shown to provide an estimate of the best achievable error of our method. It is noted that the best overall configuration in term of mean error is using intensities loss only with normal loss improving performance in some objects and decreasing in some others.

mal estimation networks can be robust to miss-calibration though data augmentation (e.g. PX-Net [21]) or be outright self-calibrating (e.g. [2]) and do not suffer from the aforementioned issue.

**LUCES-ST experiments.** LUCES-ST experiments are shown quantitatively in Table 2 and qualitatively in Figure 4. Similar to DiLiGenT-MV [19] experiments, the use of intensity rendering is also ablated an in fact it does seem to have an increase of performance for most experiments.

The best configuration turns out to be *intensity only* in terms of mean error (0.91mm) with *normals + intensity* and *normals only* being slightly better in terms of median error (0.77mm vs 0.8mm). PS-NeRF [38] and Logothetis et al. [21] are slightly worse with mean errors of (1.0mm and 1.03mm) respectively. We note that all methods in this dataset are using the single -view normal predictions from [21] (even [7] was ran with 2 epochs of normal loss only for initialisation) therefore the spread of results is much less that in DiLiGenT-MV. In addition, since accurate, near-field light calibration is available, intensity rendering improves performance on most experiments. A notable exception is the metallic *Bell* which contains environment reflections which are not modelled in the assumed rendering process.

To further re-enforce the usefulness of intensity rendering, we also add Blender renderings of the same objects with a reasonable guess of their materials (metallic *Bell*, ceramic *Owl*, porcelain *Squirrel*, etc). Note that the poses and segmentation masks are different therefore synthetic to real comparison is not fair. Nevertheless, intensity offers a clear advantage for all objects and performs the best on average with a significant margin (0.57mm intensity only vs 0.87mm normals only vs 0.75mm combined). In the real experiments, the superiority of intensity rendering is not always true as real data may contain totally un-modelled effects such as ambient light ( [24]), camera noise, and even a small calibration error. Normal estimation networks are very robust to a lot of effects and thus offer a useful source of information for real world experiments.

An additional considerations is that DiLiGenT-MV [19] contains 96 lights whereas LUCES-Stereo only uses 15. This gives a significant advantage to normal estimation networks that can use all of the lights to gain robustness to real world imperfections. In contrast, averaging the rendering loss (even L1) over a set of lights is more susceptible to clear outliers (and in fact the more lights, the higher the chance that one of the lights contains un-modelled effects such as self reflections).

Finally, to calibrate the limit of precision of out method, results with ground truth normals are also included for LUCES-Stereo (line 7 in Table 2). The obtained error of 0.33mm is much smaller than any real experiment (0.91mm) but certainly non-negligible, potentially signifying the need for higher learning capacity network.

We note that some additional visualisations including re-renderings and recovered BRDFs LUCES-Stereo experiments are available in the supplementary.

## 6. Conclusion

In this work we propose a novel neural heightmap approach to Binocular Photometric Stereo along with a new dataset - LUCES-Stereo. We show that our approach is able to extract accurate shape from extremely sparse views (i.e. 2 views) significantly better than single view photometric stereo [21] and even reach similar performance in terms of average shape error to the state-of-the-art multi-view photometric stereo method [38] on DiLiGeNT [19] benchmark.

## References

[1] Daniel G. Aliaga and Yi Xu. Photogeometric structured light: A self-calibrating and multi-viewpoint framework for accurate 3d modeling. In *CVPR*, 2008. 2

[2] Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee K. Wong. Sdps-net: Self-calibrating deep photometric stereo networks. In *CVPR*, 2019. 2, 6, 8

[3] Hao Du, Dan B. Goldman, and Steven M. Seitz. Binocular photometric stereo. In *BMVC*, 2011. 2

[4] Robert Easdon. Ambient occlusion and shadows for molecular graphics. 2013. 5

[5] Carlos Hernández Esteban, George Vogiatzis, and Roberto Cipolla. Multiview photometric stereo. *PAMI*, 2008. 2

[6] Dan B. Goldman, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and spatially-varying brdfs from photometric stereo. *PAMI*, 2010. 2

[7] Heng Guo, Hiroaki Santo, Boxin Shi, and Yasuyuki Matsushita. Edge-preserving near-light photometric stereo with neural surfaces. *arXiv*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[8] Clément Hardy, Yvain Quéau, and David Tschumperlé. Ms-ps: A multi-scale network for photometric stereo with a new comprehensive training dataset. *arXiv*, 2022. 1

[9] S. Ikehata. Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In *ECCV*, 2018. 1, 2

[10] Yakun Ju, Cong Zhang, Songsong Huang, Yuan Rao, and Kin-Man Lam. Learning deep photometric stereo network with reflectance priors. In *ICME*, 2023. 2

[11] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncertainty-aware deep multi-view photometric stereo. In *CVPR*, 2022. 2

[12] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Multi-view photometric stereo revisited. In *WACV*, 2023. 2

[13] Berk Kaya, Suryansh Kumar, Francesco Sarno, Vittorio Ferrari, and Luc Van Gool. Neural radiance fields approach to deep multi-view photometric stereo. In *WACV*, 2021. 2

[14] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Eurographics symposium on Geometry processing*, 2006. 7

[15] Hui Kong, Pengfei Xu, and Eam Khwang Teoh. Binocular uncalibrated photometric stereo. In *ISCV*, 2006. 2

[16] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean E. Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project: 3d scanning of large statues. In *SIGGRAPH*, 2000. 2

[17] Junxuan Li and Hongdong Li. Neural reflectance for shape recovery with shadow handling. In *CVPR*, pages 16221–16230, 2022. 2

[18] Junxuan Li and Hongdong Li. Self-calibrating photometric stereo by neural inverse rendering. In *ECCV*. Springer, 2022. 2

[19] Min Li, Zhenglong Zhou, Zhe Wu, Boxin Shi, Changyu Diao, and Ping Tan. Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE Trans. Image Process.*, 2020. 1, 2, 5, 6, 7, 8

[20] Fotios. Logothetis, Ignas. Budvytis, Roberto. Mecca, and Roberto. Cipolla. PX-NET: Simple, Efficient Pixel-Wise Training of Photometric Stereo Networks. In *ICCV*, 2021. 2, 4

[21] Fotios Logothetis, Roberto Mecca, Ignas Budvytis, and Roberto Cipolla. A cnn based approach for the point-light photometric stereo problem. *IJCV*, 2022. 1, 2, 3, 5, 6, 7, 8

[22] Fotios Logothetis, Roberto Mecca, and Roberto Cipolla. Semi-calibrated near field photometric stereo. In *CVPR*, 2017. 2

[23] Fotios Logothetis, Roberto Mecca, and Roberto Cipolla. A differential volumetric approach to multi-view photometric stereo. In *ICCV*, 2019. 2

[24] Fotios Logothetis, Roberto Mecca, Yvain Quéau, and Roberto Cipolla. Near-field photometric stereo in ambient light. In *BMVC*, 2016. 2, 8

[25] W. Matusik, H. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. *ACM TOG*, 2003. 4

[26] Roberto Mecca, Fotios Logothetis, Ignas Budvytis, and Roberto Cipolla. Luces: A dataset for near-field point light source photometric stereo. In *BMVC*, 2021. 2

[27] Roberto Mecca, Yvain Quéau, Fotios Logothetis, and Roberto Cipolla. A single-lobe photometric stereo approach for heterogeneous material. *SIAM Journal on Imaging Sciences*, 2016. 2

[28] Roberto Mecca, A. Wetzler, A. Bruckstein, and R. Kimmel. Near Field Photometric Stereo with Point Light Sources. *SIAM Journal on Imaging Sciences*, 2014. 4

[29] 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 *ECCV*, 2020. 1, 2

[30] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *ACM Trans. Graph.*, 2005. 2

[31] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 2013. 2

[32] Y. Quéau and J.-D. Durou. Edge-preserving integration of a normal field: Weighted least squares, TV and L1 approaches. In *SSVM*, 2015. 1

[33] Szymon Rusinkiewicz, Olaf A. Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. *ACM Trans. Graph.*, 2002. 2

[34] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[35] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 1, 2, 3

[36] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022. 2

[37] Chaoyang Wang, Lijuan Wang, Yasuyuki Matsushita, Bojun Huang, Magnetro Chen, and Frank K. Soong. Binocular photometric stereo acquisition and reconstruction for 3d talking head applications. In *INTERSPEECH*, 2013. 2

[38] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In *ECCV*, 2022. 1, 2, 5, 6, 7, 8

[39] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. S3-NeRF: Neural reflectance field from shading and shadow under a single viewpoint. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *NeurIPS*, 2022. 2

[40] Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graph.*, 2004. 2

[41] Michael Zollhöfer, Angela Dai, Matthias Innmann, Chenglei Wu, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Shading-based refinement on volumetric signed distance functions. *ACM Trans. Graph.*, 2015. 2