Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images

Sai Bi[1] Zexiang Xu[1] Kalyan Sunkavalli[2] David Kriegman[1] Ravi Ramamoorthi[1]
[1]UC San Diego [2]Adobe Research

Abstract

We introduce a novel learning-based method to reconstruct the high-quality geometry and complex, spatially-varying BRDF of an arbitrary object from a sparse set of only six images captured by wide-baseline cameras under collocated point lighting. We first estimate per-view depth maps using a deep multi-view stereo network; these depth maps are used to coarsely align the different views. We propose a novel multi-view reflectance estimation network architecture that is trained to pool features from these coarsely aligned images and predict per-view spatially-varying diffuse albedo, surface normals, specular roughness and specular albedo. Finally, we fuse and refine these per-view estimates to construct high-quality geometry and per-vertex BRDFs. We do this by jointly optimizing the latent space of our multi-view reflectance network to minimize the photometric error between images rendered with our predictions and the input images. While previous state-of-the-art methods fail on such sparse acquisition setups, we demonstrate, via extensive experiments on synthetic and real data, that our method produces high-quality reconstructions that can be used to render photorealistic images.

1. Introduction

Reconstructing the 3D geometry and reflectance properties of an object from 2D images has been a long-standing problem in computer vision and graphics, with applications including 3D visualization, relighting, and augmented and virtual reality. Traditionally this has been accomplished using complex acquisition systems [5, 17, 41, 45, 56] or multi-view stereo (MVS) methods [13, 40] applied to dense image sets [34, 49]. The acquisition requirements for these methods significantly limits their practicality. Recently, deep neural networks have been proposed for material estimation from a single or a few images. However, many of these methods are restricted to estimating the spatially-varying BRDF (SVBRDF) of planar samples [10, 15, 31]. Li et al. [32] demonstrate shape and reflectance reconstruction from a single image, but their reconstruction quality is limited by their single image input.

Our goal is to enable practical *and* high-quality shape and appearance acquisition. To this end, we propose using a simple capture setup: a sparse set of six cameras—placed at one vertex and the centers of the adjoining faces of a regular icosahedron, forming a 60° cone—with collocated point lighting (Fig. 2 left). Capturing six images should allow for better reconstruction compared to single image methods. However, at such wide baselines, the captured images have few correspondences and severe occlusions, making it challenging to fuse information across viewpoints.

Figure 1: From six wide-baseline input images of an object captured under collocated point lighting (top row), our method reconstructs high-quality geometry and spatially-varying, non-Lambertian reflectance (bottom row, a tone mapping is performed on specular albedo to make it more visible), allowing us to re-render the captured object under novel viewpoint and illumination (bottom, right).

As illustrated in Fig. 2, we propose a two-stage approach to address this problem. First, we design **multi-view geometry and reflectance estimation networks that regress the 2D depth, normals and reflectance for each input view by robustly aggregating information across all sparse viewpoints.** We estimate the depth for each input view using a deep multi-view stereo network [50, 53] (Sec. 3.1). Because of our sparse capture, these depth maps contain errors and cannot be used to accurately align the images to estimate per-vertex BRDFs [34, 56]. Instead, we use these depth maps to warp the images to one viewpoint and use a *novel deep multi-view reflectance estimation network* to estimate per-pixel normals and reflectance (parameterized by diffuse albedo, specular albedo and roughness in a simplified Disney BRDF model [24]) for that viewpoint (Sec. 3.2). This network extracts features from the warped images, aggregates

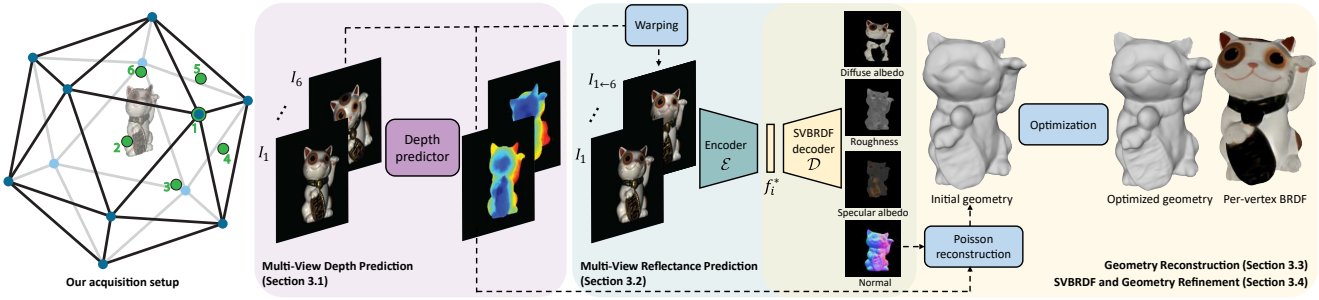

Figure 2: Our acquisition setup (leftmost figure) and framework. We capture six images with collocated cameras and lights placed at a vertex (*green* circle 1) and five adjoining face centers (*green* circle 2-6) of an icosahedron. Using the six images, we predict per-view depth (*red block*). We warp the input images using the predicted depths and pass them to a multi-view SVBRDF estimation network to get per-view SVBRDFs (*blue block*). Finally, we reconstruct 3D geometry from the estimated depth and normals, and perform a joint optimization to get refined geometry and per-vertex BRDFs (*yellow block*).

them across viewpoints using max-pooling, and decodes the pooled features to estimate the normals and SVBRDF for that viewpoint. This approach to aggregate multi-view information leads to more robust reconstruction than baseline approaches like a U-Net architecture [38], and we use it to recover normals and reflectance for each view.

Second, we propose a **novel method to fuse these per-view estimates into a single mesh with per-vertex BRDFs using optimization in a learnt reflectance space**. First, we use Poisson reconstruction [25] to construct a mesh from the estimated per-view depth and normal maps (Sec. 3.3). Each mesh vertex has multiple reflectance parameters corresponding to each per-view reflectance map, and we fuse these estimates to reconstruct object geometry and reflectance that will *accurately reproduce the input images*. Instead of optimizing the per-vertex reflectance parameters, which leads to outliers and spatial discontinuities, we optimize the *the latent features of our multi-view reflectance estimation network* (Sec. 3.4). We pass these latent features to the reflectance decoder to construct per-view SVBRDFs, fuse them using per-vertex blending weights, and render them to compute the photometric error for all views. This entire pipeline is differentiable, allowing us to backpropagate this error and iteratively update the reflectance latent features and per-vertex weights till convergence. This process refines the reconstruction to best match the specific captured images, while leveraging the priors learnt by our reflectance estimation network.

We train our networks with a large-scale synthetic dataset comprised of procedurally generated shapes with complex SVBRDFs [50, 52] and rendered using a physically-based renderer. While our method is trained with purely synthetic data, it generalizes well to real scenes. This is illustrated in Figs. 1 and 8, where we are able to reconstruct real objects with complex geometry and non-Lambertian reflectance. Previous state-of-the-art methods, when applied to sparse input images for such objects, produce incomplete, noisy geometry and erroneous reflectance estimates (Figs. 4 and 7). In contrast, our work is the first to reconstruct detailed geometry and high-quality reflectance from sparse multi-view inputs, allowing us to render photorealistic images under novel view and lighting.

2. Related Works

3D reconstruction. To reconstruct 3D geometry from image sets, traditional methods [14, 28, 40] find correspondences between two or more images utilizing specific image features. Such methods are sensitive to illumination changes, non-Lambertian reflectance and textureless surfaces. The existence of multiple points with similar matching costs also require these methods to have a large number of images to get high-quality reconstructions (we refer the interested readers to [14] for more details). In contrast, our method reconstructs high-quality geometry for complex real scenes from an order of magnitude fewer images.

Recently, numerous learning-based methods have been proposed to reconstruct 3D shape using various geometric representations, including regular volumes [21, 36, 47], point clouds [1, 42] and depth maps [18, 53]. These methods cannot produce high-resolution 3D meshes. We extend recent learning-based MVS frameworks [50, 53] to estimate depth from sparse multi-view images of objects with complex reflectance. We combine this depth with estimated surface normals to reconstruct 3D meshes with fine details.

SVBRDF acquisition. SVBRDF acquisition is a challenging task that often requires a dense input image set [12, 34, 49]. Many methods utilize sophisticated hardware [33] or light patterns [17, 22, 41]. Reconstruction from sparse images has been demonstrated for planar objects [3, 31, 51], and known geometry [56]. In contrast, we reconstruct the geometry and complex reflectance of arbitrary objects from a sparse set of six input images.

Photometric stereo methods have been proposed to reconstruct arbitrary shape and SVBRDFs [4, 16]; however, they focus on single-view reconstruction and require hundreds

of images. Recent works [19, 34] utilize images captured by a collocated camera-light setup for shape and SVBRDF estimation. In particular, Nam et al. [34] capture more than sixty images and use multi-view reconstruction and physics-based optimization to recover geometry and reflectance. In contrast, by designing novel deep networks, we are able to reconstruct objects from only six images.

Learning-based methods have been applied for normal and SVBRDF acquisition. Deep photometric stereo methods reconstruct surface normals from tens to hundreds of images [6, 7] but they do not address reflectance or 3D geometry estimation. Most deep SVBRDF acquisition methods are designed for planar samples [2, 10, 11, 15, 30, 31]. Some recent multi-image SVBRDF estimation approaches pool latent features from multiple views [11] and use latent feature optimization [15] but they only handle planar objects. Li et al. [32] predict depth and SVBRDF from a single image; however, a single input does not provide enough information to accurately reconstruct geometry and reflectance. By capturing just six images, our approach generates significantly higher quality results.

3. Algorithm

Our goal is to accurately reconstruct the geometry and SVBRDF of an object with a simple acquisition setup. Recent work has utilized collocated point illumination for reflectance estimation from a sparse set of images [2, 3, 10, 31]; such lighting minimizes shadows and induces high-frequency effects like specularities, making reflectance estimation easier. Similarly, Xu et al. [50] demonstrate novel view synthesis from sparse multi-view images of a scene captured under a single point light.

Motivated by this, we utilize a similar capture system as Xu et al.—six cameras placed at one vertex of a regular icosahedron and the centers of the five faces adjoining that vertex. Unlike their use of a single point light for all images, we capture each image under a point light (nearly) collocated with the corresponding camera (see Fig. 2 left). The setup is calibrated giving us a set of $n = 6$ input images, $\{I_i\}_{i=1}^n$ with the corresponding camera calibration. This wide baseline setup—with an angle of 37° between the center and boundary views—makes it possible to image the entire object with a small set of cameras. In the following, we describe how we reconstruct an object from these sparse input images.

3.1. Multi-View Depth Prediction

Traditional MVS methods depend on hand-crafted features such as Harris descriptors to find correspondence between views. Such features are not robust to illumination changes or non-Lambertian surfaces, making them unusable for our purposes. In addition, due to the sparse inputs and large baselines, parts of the object may be visible in as few as two views. These factors cause traditional MVS methods to fail at finding accurate correspondences, and thus fail to reconstruct high-quality geometry.

Instead, we make use of a learning-based method to estimate the depth. Given the input images $\{I_i\}_{i=1}^n$, we estimate the depth map D_i for view i. Similar to recent works on learning-based MVS [20, 50, 53], our network consists of two components: a feature extractor $\mathcal{F}$ and a correspondence predictor $\mathcal{C}$. The feature extractor is a 2D U-Net [38] that extracts a 16-channel feature map for each image I_i. To estimate the depth map at I_i, we warp the feature maps of all views to view i using a set of 128 pre-defined depth levels, and build a 3D plane sweep volume [9] by calculating the variance of feature maps over views. The 3D volume is further fed to the correspondence predictor $\mathcal{C}$ that is a 3D U-Net to predict the probability of each depth level. We calculate the depth as a probability-weighted sum of all depth levels. The training loss is defined as the L_1 loss between predicted depths and ground truth depths. By learning the feature representations and correspondence, the proposed framework is more robust to illumination changes and specularities, thus producing more accurate pixel-wise depth predictions than traditional methods.

While such networks are able to produce reasonable depth, the recovered depth has errors in textureless regions. To further improve the accuracy, we add a guided filter module [46] to the network, which includes a guided map extractor $\mathcal{G}$ as well as a guided layer g. Let the initial depth prediction at view i be D_i'. The guided map extractor $\mathcal{G}$ takes image I_i as input and learns a guidance map $\mathcal{G}(I_i)$. The final depth map is estimated as:

$$D_i = g(\mathcal{G}(I_i), D_i'). \tag{1}$$

The training loss is defined as the L_1 distance between predicted depths and ground truth depths. All components are trained jointly in an end-to-end manner.

3.2. Multi-View Reflectance Prediction

Estimating surface reflectance from sparse images is a highly under-constrained problem. Previous methods either assume geometry is known [2, 3, 31, 10] or can be reconstructed with specific devices [17] or MVS [34]. In our case, accurate geometry cannot be reconstructed from sparse inputs with traditional MVS methods. While our learning-based MVS method produces reasonable depth maps, they too have errors, making it challenging to use them to align the images and estimate per-vertex SVBRDF. Instead, for each input image I_i, we first estimate its corresponding normals, I_i, and SVBRDF, represented by diffuse albedo A_i, specular roughness R_i and specular albedo S_i.

To estimate the SVBRDF at view i, we warp all input images to this view using predicted depths D_i. One approach for multi-view SVBRDF estimation could be to feed

this stack of warped images to a convolutional neural network like the commonly used U-Net [31, 38]. However, the inaccuracies in the depth maps lead to misalignments in the warped images, especially in occluded regions, and this architecture is not robust to these issues.

We propose a novel architecture that is robust to depth inaccuracies and occlusions. As shown in Fig. 3, our network comprises a Siamese encoder [8], $\mathcal{E}$, and a decoder, $\mathcal{D}$, with four branches for the four SVBRDF components. To estimate the SVBRDF at a reference view i, the encoder processes n pairs of inputs, each pair including image I_i as well as the warped image $I_{i\leftarrow j}$, where we warp image I_j at view j to the reference view i using the predicted depth D_i. To handle potential occlusions, directly locating occluded regions in the warped images using predicted depths and masking them out is often not feasible due to inaccurate depths. Instead we keep the occluded regions in the warped images and include the depth information in the inputs, allowing the network to learn which parts are occluded.

To include the depth information, we draw inspiration from the commonly used shadow mapping technique [44]. The depth input consists of two components: for each pixel in view i, we calculate its depths $Z_{i\leftarrow j}$ in view j; we also sample its depth $Z^*_{i\leftarrow j}$ from the depth map D_j by finding its projections on view j. Intuitively if $Z_{i\leftarrow j}$ is larger than $Z^*_{i\leftarrow j}$, then the pixel is occluded in view j; otherwise it is not occluded. In addition, for each pixel in the reference view i, we also include the lighting directions L_i of the light at view i, as well as the lighting direction of the light at view j, denoted as $L_{i\leftarrow j}$. We assume a *point light* model here. Since the light is collocated with the camera, by including the lighting direction we are also including the viewing direction of each pixel in the inputs. All directions are in the coordinate system of the reference view. Such cues are critical for networks to infer surface normals using photometric information. Therefore, the input for a pair of views i and j is:

$$H_{i,j} = \{I_i, I_{i\leftarrow j}, Z_{i\leftarrow j}, Z^*_{i\leftarrow j}, L_i, L_{i\leftarrow j}\}. \tag{2}$$

The input contains 14 channels in total, and there are a total of n such inputs. We feed all the inputs to the encoder network $\mathcal{E}$ and get the intermediate features $f_{i,j}$. All these intermediate features are aggregated with a max-pooling layer yielding a common feature representation for view i, f^*_i:

$$f_{i,j} = \mathcal{E}(H_{i,j}) \tag{3}$$

$$f^*_i = \text{max-pool}(\{f_{i,j}\}_{j=1}^{n}) \tag{4}$$

f^*_i is fed to the decoder to predict each SVBRDF component for view i:

$$A_i, N_i, R_i, S_i = \mathcal{D}(f^*_i) \tag{5}$$

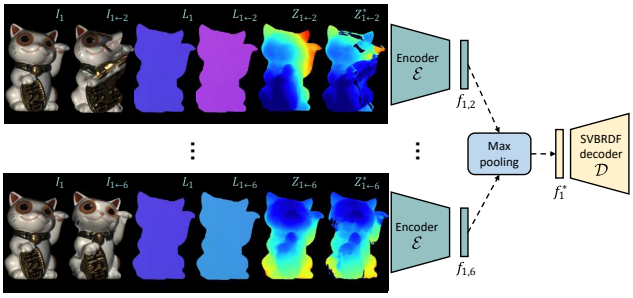

Figure 3: Our multi-view SVBRDF estimation network. An encoder extracts features from reference and warped image pairs. These features are max-pooled to get a single reference-view feature map, which is decoded to predict that view's SVBRDF. Note the errors in the warped images; max-pooling mitigates their effect on the output SVBRDF.

Compared to directly stacking all warped images together, our proposed network architecture works on pairs of input images and aggregates features across views using a max-pooling layer. The use of max-pooling makes the network more robust to occlusions and misalignments caused by depth inaccuracies and produces more accurate results (see Tab. 1). It also makes the network invariant to the number and order of the input views, a fact that could be utilized for unstructured capture setups. The training loss $\mathcal{L}$ of the network is defined as:

$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_N + \mathcal{L}_R + \mathcal{L}_S + \mathcal{L}_I \tag{6}$$

where the first four terms are the L_2 losses for each SVBRDF component, and $\mathcal{L}_I$ is the L_2 loss between input images and rendered images generated with our predictions.

3.3. Geometry Reconstruction

The previous multi-view depth and SVBRDF estimation networks give us per-view depth and normal maps at full-pixel resolution. We fuse these per-view estimates to reconstruct a single 3D geometry for the object. We first build a point cloud from the depth maps, by generating 3D points from each pixel in every per-view depth map. For each point, we also get its corresponding normal from the estimated normal maps. Given this set of 3D points with surface normals, we perform a Poisson reconstruction [26] to reconstruct the fused 3D geometry. The initial point clouds may contain outliers due to inaccuracies in the depth maps. To get rid of undesired structures in the output geometry, we generate a coarse initial geometry by setting the depth of the spatial octree in Poisson reconstruction to 7—corresponding to an effective voxel resolution of 128^3. We refine this initial geometry in the subsequent stage. Compared to learning-based 3D reconstruction methods that directly generate geometry (voxel grids [23, 37], implicit functions [35, 39] or triangle meshes [43]) from images, this approach generalizes to arbitrary shapes and produces more detailed reconstructions.

3.4. SVBRDF and Geometry Refinement

Given the initial coarse geometry as well as the per-view SVBRDF predictions, we aim to construct a detailed 3D mesh with per-vertex BRDFs. For each vertex, a trivial way to get its BRDF is to blend the predicted SVBRDFs across views using pre-defined weights such as the dot product of the viewing directions and surface normals. However, this leads to blurry results (Fig. 5), due to the inconsistencies in the estimated SVBRDFs and the geometry. Also note that our SVBRDF predictions are computed from a single feed-forward network pass, and are not guaranteed to reproduce the captured input images exactly because the network has been trained to minimize the reconstruction loss on the entire training set and not this specific input sample.

We address these two issues with a novel rendering-based optimization that estimates per-vertex BRDFs that minimize the error between rendering the predicted parameters and the captured images. Because of the sparse observations, independently optimizing per-vertex BRDFs leads to artifacts such as outliers and spatial discontinuities, as shown in Fig. 5. Classic inverse rendering methods address this using hand-crafted priors. Instead, we optimize the per-view feature maps f_i^* that are initially predicted from our SVBRDF encoder (Eqn. 4). These latent features, by virtue of the training process, capture the manifold of object reflectances, and generate spatially coherent per-view SVBRDFs when passed through the decoder, $\mathcal{D}$ (Eqn. 5). *Optimizing in this feature space allows us to adapt the reconstruction to the input images, while leveraging the priors learnt by our multi-view SVBRDF estimation network.*

Per-vertex BRDF and color. For each vertex v_k, we represent its BRDF b_k as a weighted average of the BRDF predictions from multiple views:

$$b_k = \sum_{i=1}^{n} w_{k,i} \mathcal{D}(p_{k,i}; f_i^*), \tag{7}$$

where $p_{k,i}$ is the corresponding pixel position of v_k at view i, $\mathcal{D}(p_{k,i}; f_i^*)$ represents the SVBRDF prediction at $p_{k,i}$ from view i by processing f_i^* via the decoder network $\mathcal{D}$, and $w_{k,i}$ are the per-vertex view blending weights. The rendered color of v_k at view i is calculated as:

$$I_i^*(p_{k,i}) = \Theta(b_k, L_i(p_{k,i})), \tag{8}$$

where $L_i(p_{k,i})$ is the lighting direction and also the viewing direction of vertex v_k at view i, and Θ is the rendering equation. We assume a point light source collocated with the camera (which allows us to ignore shadows), and only consider direct illumination in the rendering equation.

Per-view warping. Vertex v_k can be projected onto view i using the camera calibration; we refer to this projection as $u_{k,i}$. However, the pixel projections onto multiple views might be inconsistent due to inaccuracies in the reconstructed geometry. Inspired by Zhou et al. [55], we apply a non-rigid warping to each view to better align the projections. In particular, for each input view, we use a $T \times T$ grid with $C = T^2$ control points ($T = 11$ in our experiments) to construct a smooth warping field over the image plane. Let $t_{i,c}$ be the translation vectors of control points at view i. The resulting pixel projection, $p_{k,i}$, is given by:

$$p_{k,i} = u_{k,i} + \sum_{c=1}^{C} \theta_c(u_{k,i}) t_{i,c}, \tag{9}$$

where θ_c returns the bilinear weight for a control point $t_{i,c}$ at pixel location $u_{k,i}$.

SVBRDF optimization. We optimize per-view latent features f_i^*, per-vertex blending weights $w_{k,i}$ and per-view warping fields $t_{i,c}$ to reconstruct the final SVBRDFs. The photometric consistency loss between the rendered colors and ground truth colors for all K vertices is given by:

$$E_{\text{photo}}(f_i^*, w, t) = \frac{1}{n \cdot K} \sum_{k=1}^{K} \sum_{i=1}^{n} ||I_i^*(p_{k,i}) - I_i(p_{k,i})||_2^2.$$

We clamp the rendered colors to the range of $[0, 1]$ before calculating the loss. To prevent the non-rigid warping from drifting, we also add an L_2 regularizer to penalize the norm of the translation vectors:

$$E_{\text{warp}}(t) = \frac{1}{n \cdot C} \sum_{i=1}^{n} \sum_{c=1}^{C} ||t_{i,c}||_2^2. \tag{10}$$

Therefore the final energy function for the optimization is:

$$E = E_{\text{photo}}(f^*, w, t) + \lambda E_{\text{warp}}(t). \tag{11}$$

We set λ to 100 , and optimize the energy function with Adam optimizer [27] with a learning rate of 0.001.

Geometry optimization. We use the optimized per-vertex normal, n_k, to update the geometry of the object by re-solving the Poisson equation (Sec. 3.3). Unlike the initial geometry reconstruction, we set the depth of the spatial octree to 9—corresponding to a voxel resolution of 512^3—to better capture fine-grained details of the object. We use this updated geometry in subsequent SVBRDF optimization iterations. We update the geometry once for every 50 iterations of SVBRDF optimization, and we perform $400 - 1000$ iterations for the SVBRDF optimization.

Per-vertex refinement. The bottleneck in our multi-view SVBRDF network—that we use as our reflectance representation—may cause a loss of high-frequency details in the predicted SVBRDFs. We retrieve these details back

by directly optimizing the BRDF parameters b_k of each vertex to minimizing the photometric loss in Eqn. (10). Note that after the previous optimization, the estimated BRDFs have already converged to good results and the rendered images are very close to the input images. Therefore, in this stage, we use a small learning rate (0.0005), and perform the optimization for a small number ($40 - 100$) of iterations.

4. Implementation and Results

Training data. We follow Xu et al. [50] and procedurally generate complex scenes by combining 1 to 5 primitive shapes such as cylinders and cubes displaced by random height maps. We generate $20,000$ training and 400 testing scenes. We divide the high-quality materials from the Adobe Stock dataset[1] into a training and testing set, and use them to texture the generated scenes separately. For each scene, following the setup discussed in Sec. 1, we render the 6 input view images with a resolution of 512×512 using a custom Optix-based global illumination renderer with 1000 samples per pixel. We also render the ground truth depth, normals, and SVBRDF components for each view.

Network architecture. For depth estimation, we use a 2D U-Net architecture [38] for the feature extractor, $\mathcal{F}$, and guidance map extractor, $\mathcal{G}$. Both networks have 2 downsampling/upsampling blocks. The correspondence predictor $\mathcal{C}$ is a 3D U-Net with 4 downsampling/upsampling blocks. For multi-view SVBRDF estimation, both the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ are 2D CNNs, with 3 downsampling layers in $\mathcal{E}$ and 3 upsampling layers in $\mathcal{D}$. Note that we *do not* use skip connections in the SVBRDF network; this forces the latent feature to learn a meaningful reflectance space and allows us to optimize it in our refinement step. We use group normalization [48] in all networks. We use a differentiable rendering layer that computes local shading under point lighting without considering visibility or global illumination. This is a reasonable approximation in our collocated lighting setup. For more details, please refer to the supplementary document.

Training details. All the networks are trained with the Adam optimizer [27] for 50 epochs with a learning rate of 0.0002. The depth estimation networks are trained on cropped patches of 64×64 with a batch size of 12, and the SVBRDF estimation networks are trained on cropped 320×320 patches with a batch size of 8. Training took around four days on 4 NVIDIA Titan 2080Ti GPUs.

Run-time. Our implementaion has not been optimized for the best timing efficiency. In practice, our method takes around 15 minutes for full reconstruction from images with a resolution of 512×512, where most of the time is for geometry fusion and optimization.

[1] https://stock.adobe.com/search/3d-assets

	Diffuse	Normal	Roughness	Specular
Naive U-Net	**0.0060**	0.0336	0.0359	0.0125
Ours	0.0061	**0.0304**	**0.0275**	**0.0086**
Li et al. [32]	0.0227	0.1075	0.0661	—
Ours (256×256)	**0.0047**	**0.0226**	**0.0257**	**0.0083**

Table 1: Quantitative SVBRDF evaluation on a synthetic test set. We report the L_2 error. Since Li et al. [32] work on 256×256 images, we downsample and evaluate at that resolution. Also, they do not predict the specular albedo.

4.1. Evaluation on Synthetic Data

We evaluate our max-pooling-based multi-view SVBRDF estimation network on our synthetic test set. In particular, we compare it with a baseline U-Net (with 5 downsampling/upsampling blocks) that takes a stack of all the coarsely aligned images ($H_{i,j} \forall j$ in Eqn. 2) as input for its encoder, and skip connections from the encoder to the four SVBRDF decoders. This architecture has been widely used for SVBRDF estimation [10, 31, 32]. As can be seen in Tab. 1, while our diffuse albedo prediction is slightly (1.7%) worse than the U-Net we significantly outperform it in specular albedo, roughness and normal predictions, with 31%, 23% and 9.5% lower L_2 loss respectively. This is in spite of not using skip-connections in our network (to allow for optimization later in our pipeline). We also compare our results with the state-of-the-art single-image shape and SVBRDF estimation method of Li et al. [32]. Unsurprisingly, we outperform them significantly, demonstrating the usefulness of aggregating multi-view information.

4.2. Evaluation on Real Captured Data

We evaluate our method on real data captured using a gantry with a FLIR camera and a nearly collocated light to mimic our capture setup. **Please refer to the supplementary material for additional results.**

Evaluation of geometry reconstruction. Our framework combines our predicted depths and normals to reconstruct the initial mesh. Figure 4 shows the comparison between our reconstructed mesh and the mesh from COLMAP, a state-of-the-art multi-view stereo framework [40]. From such sparse inputs and low-texture surfaces, COLMAP is not able to find reliable correspondence across views, which results in a noisy, incomplete 3D mesh. In contrast, our initial mesh is already more complete and detailed, as a result of our more accurate depths and normals. Our joint optimization further refines the per-vertex normals and extracts fine-scale detail in the object geometry.

Evaluation of SVBRDF optimization. We compare our SVBRDF and geometry optimization scheme (Sec. 3.4) with averaging the per-view predictions using weights based on

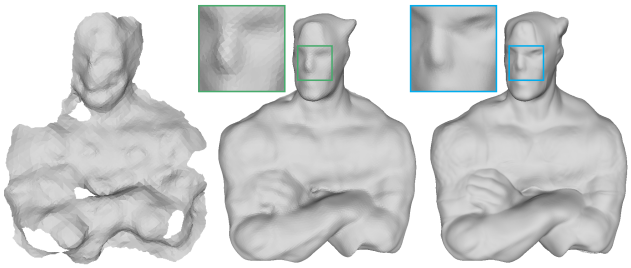

Figure 4: Comparison on geometry reconstruction. COLMAP fails to reconstruct a complete mesh from the sparse inputs. In contrast, our initial mesh is of much higher quality, and our joint optimization recovers even more fine-grained details on the mesh. Input image in Fig. 8 (top).

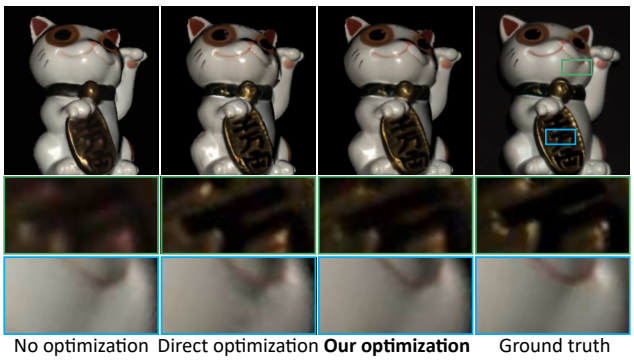

Figure 5: Comparison on SVBRDF optimization. Simple averaging without optimization produces blurry results, and direct per-vertex optimization results in outliers and discontinuities. In comparison, our optimization generates more visually plausible results.

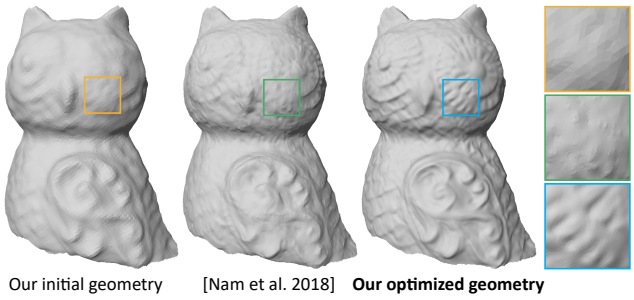

Figure 6: Comparison with Nam et al. [34]. While both have the same initialization, our learning-based refinement produces more accurate, detailed geometry. Input in Fig. 8.

the angle between the viewpoint and surface normal, as well as this averaging followed by per-vertex optimization. From Fig. 5 we can see that the weighted averaging produces blurry results. Optimizing the per-vertex BRDFs brings back detail but also has spurious discontinuities in appearance because of the lack of any regularization. In contrast, our latent-space optimization method recovers detailed appearance without these artifacts.

Comparisons against Nam et al. [34] We also compare our work with the state-of-the-art geometry and reflectance reconstruction method of Nam et al. Their work captures 60+ images of an object with a handheld camera under collocated lighting; they first use COLMAP [40] to reconstruct the coarse shape and use it to bootstrap a physics-based optimization process to recover per-vertex normals and BRDFs. COLMAP cannot generate complete meshes from our sparse inputs (see Fig. 4). Therefore, we provided our input images, camera calibration, *and initial geometry* to the authors who processed this data. As can be seen in Fig. 6, our final reconstructed geometry has significantly more details than their final optimized result in spite of starting from the same initialization. Since they use a different BRDF representation than ours, making direct SVBRDF comparisons difficult, in Fig. 7 we compare renderings of the reconstructed object under novel lighting and viewpoint. These results show that they cannot handle our sparse input and produce noise, erroneous reflectance (CAT scene) or are unable to recover the specular highlights of highly specular objects (CACTUS) scene. In comparison, our results have significantly higher visual fidelity. Please refer to the supplementary video for more renderings.

More results on real data. Figure 8 shows results from our method on additional real scenes. We can see here that our method can reconstruct detailed geometry and appearance for objects with a wide variety of complex shapes and reflectance. Comparing renderings of our estimates under novel camera and collocated lighting against ground truth captured photographs demonstrates the accuracy of our reconstructions. We can also photorealistically render these objects under novel environment illumnination. Please refer to the supplementary document and video for more results.

Limitations. Our method might fail to handle highly non-convex objects, where some parts are visible in as few as a single view and there are no correspondence cues to infer correct depth. In addition, we do not consider global illumination in SVBRDF optimization. While it is a reasonable approximation in most cases, it might fail in some particular scenes with strong inter-reflections. For future work, it would be interesting to combine our method with physics-based differentiable rendering [29, 54] to handle these complex light transport effects.

5. Conclusion

We have proposed a learning-based framework to reconstruct the geometry and appearance of an arbitrary object from a sparse set of just six images. We predict per-view depth using learning-based MVS, and design a novel multi-view reflectance estimation network that robustly aggregates information from our sparse views for accurate normal and SVBRDF estimation. We further propose a novel joint

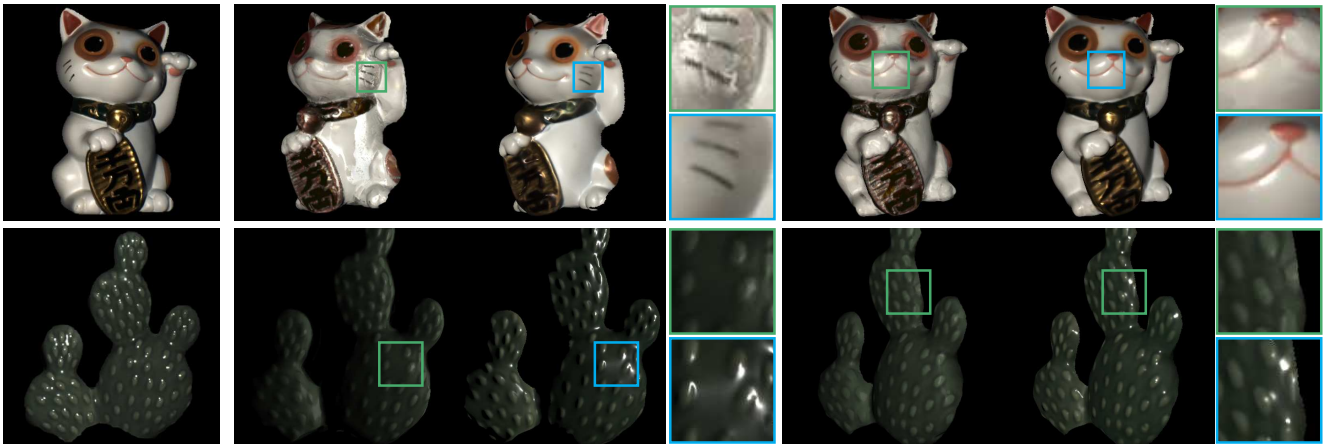

Figure 7: Comparison with Nam et al. [34]. We render two reconstructed objects under *novel* viewpoints and lighting. Nam et al. are not able to accurately reconstruct appearance from sparse views, and produce noisy edges and incorrect specular highlights (top) or miss the specular component completely (bottom). In contrast, our method produces photorealistic results.

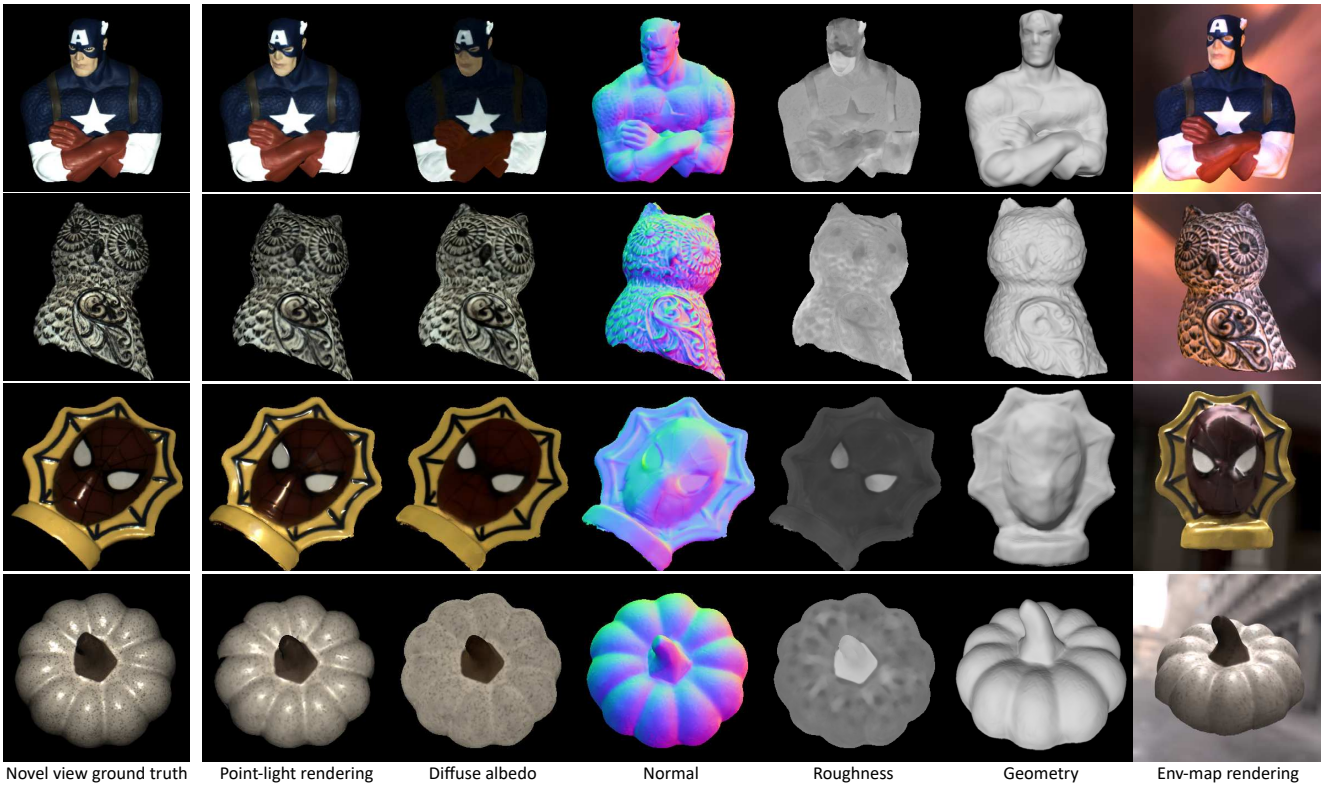

Figure 8: Results on real scenes. For each scene, we show our reconstructed geometry, normal map and SVBRDF components (please refer to supplementary materials for specular albedo). We compare our point-light rendering results (second column) under *novel* viewpoints and lighting with captured ground truth photographs (first column). We also show a rendering of the object with our reconstructed appearance under environment lighting (last column).

optimization in latent feature space to fuse and refine our multi-view predictions. Unlike previous methods that require densely sampled images, our method produces high-quality reconstructions from a sparse set of images, and presents a step towards practical appearance capture for 3D scanning and VR/AR applications.

Acknowledgements This work was supported in part by NSF grant 1617234, ONR grants N000141712687, N000141912293, Adobe and the UC San Diego Center for Visual Computing.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, pages 40–49, 2018. 2

[2] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Trans. Graph.*, 35(4):65:1–65:13, July 2016. 3

[3] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Two-shot svbrdf capture for stationary materials. *ACM Transactions on Graphics*, 34(4):110:1–110:13, July 2015. 2, 3

[4] Neil Alldrin, Todd Zickler, and David Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *CVPR*, pages 1–8. IEEE, 2008. 2

[5] Seung-Hwan Baek, Daniel S Jeon, Xin Tong, and Min H Kim. Simultaneous acquisition of polarimetric SVBRDF and normals. *ACM Transactions on Graphics*, 37(6):268–1, 2018. 1

[6] Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee K Wong. Self-calibrating deep photometric stereo networks. In *ECCV*, 2018. 3

[7] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Ps-fcn: A flexible learning framework for photometric stereo. In *ECCV*, 2018. 3

[8] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546, 2005. 4

[9] Robert T Collins. A space-sweep approach to true multi-image matching. In *CVPR*, pages 358–363. IEEE, 1996. 3

[10] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. *ACM Transactions on Graphics*, 37(4):128, 2018. 1, 3, 6

[11] Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. Flexible svbrdf capture with a multi-image deep network. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)*, 38(4), July 2019. 3

[12] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics*, 33(6):193, 2014. 2

[13] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 1

[14] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 2

[15] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Transactions on Graphics*, 38(4):134, 2019. 1, 3

[16] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2009. 2

[17] Michael Holroyd, Jason Lawrence, and Todd Zickler. A coaxial optical scanner for synchronous acquisition of 3D geometry and surface reflectance. *ACM Transactions on Graphics*, 29(4):99, 2010. 1, 2, 3

[18] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *CVPR*, pages 2821–2830, 2018. 2

[19] Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, Jian Wang, and Aswin C Sankaranarayanan. Reflectance capture using univariate sampling of brdfs. In *ICCV*, pages 5362–5370, 2017. 3

[20] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. DPSNet: end-to-end deep plane sweep stereo. *ICLR*, 2019. 3

[21] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfaceNet: An end-to-end 3D neural network for multiview stereopsis. In *ICCV*, pages 2307–2315, 2017. 2

[22] Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. Efficient reflectance capture using an autoencoder. *ACM Transactions on Graphics*, 37(4):127–1, 2018. 2

[23] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, pages 365–376, 2017. 4

[24] Brian Karis and Epic Games. Real shading in unreal engine 4. 1

[25] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2

[26] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):29, 2013. 4

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 5, 6

[28] F. Langguth, K. Sunkavalli, S. Hadap, and M. Goesele. Shading-aware multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2

[29] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 7

[30] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Trans. Graph.*, 36(4):45:1–45:11, July 2017. 3

[31] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *ECCV*, pages 72–87, 2018. 1, 2, 3, 4, 6

[32] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia 2018*, page 269. ACM, 2018. 1, 3, 6

[33] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003. 2

[34] Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. Practical SVBRDF acquisition of 3D objects with unstruc-

tured flash photography. In *SIGGRAPH Asia 2018*, page 267. ACM, 2018. 1, 2, 3, 7, 8

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 4

[36] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*, pages 1936–1944, 2018. 2

[37] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. OctnetFusion: Learning depth fusion from data. In *2017 International Conference on 3D Vision*, pages 57–66. IEEE, 2017. 4

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3, 4, 6

[39] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, October 2019. 4

[40] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 2, 6, 7

[41] Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Transactions on graphics*, 32(4):109, 2013. 1, 2

[42] Jinglu Wang, Bo Sun, and Yan Lu. Mvpnet: Multi-view point regression networks for 3d object reconstruction from a single image. *arXiv preprint arXiv:1811.09410*, 2018. 2

[43] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single rgb images. In *ECCV*, pages 52–67, 2018. 4

[44] Lance Williams. Casting curved shadows on curved surfaces. In *SIGGRAPH*, volume 12, pages 270–274. ACM, 1978. 4

[45] Hongzhi Wu, Zhaotian Wang, and Kun Zhou. Simultaneous localization and appearance estimation with a consumer rgb-d camera. *IEEE Transactions on visualization and computer graphics*, 22(8):2012–2023, 2015. 1

[46] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast end-to-end trainable guided filter. In *CVPR*, pages 1838–1847, 2018. 3

[47] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *NIPS*, pages 540–550, 2017. 2

[48] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 6

[49] Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Transactions on Graphics*, 35(6):187, 2016. 1, 2

[50] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics*, 38(4):76, 2019. 1, 2, 3, 6

[51] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. Minimal brdf sampling for two-shot near-field reflectance acquisition. *ACM Transactions on Graphics*, 35(6):188, 2016. 2

[52] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics*, 37(4):126, 2018. 2

[53] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783, 2018. 1, 2, 3

[54] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38(6), 2019. 7

[55] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3D reconstruction with consumer depth cameras. *ACM Transactions on Graphics*, 33(4):155, 2014. 5

[56] Zhiming Zhou, Guojun Chen, Yue Dong, David Wipf, Yong Yu, John Snyder, and Xin Tong. Sparse-as-possible SVBRDF acquisition. *ACM Transactions on Graphics*, 35(6):189, 2016. 1, 2