Learning Photometric Feature Transform for Free-form Object Scan

Xiang Feng, Kaizhang Kang, Fan Pei, Huakeng Ding, Jinjiang You,
Ping Tan, Kun Zhou *Fellow, IEEE* and Hongzhi Wu *Member, IEEE*

Abstract—We propose a novel framework to automatically learn to aggregate and transform photometric measurements from multiple unstructured views into spatially distinctive and view-invariant low-level features, which are subsequently fed to a multi-view stereo pipeline to enhance 3D reconstruction. The illumination conditions during acquisition and the feature transform are jointly trained on a large amount of synthetic data. We further build a system to reconstruct both the geometry and anisotropic reflectance of a variety of challenging objects from hand-held scans. The effectiveness of the system is demonstrated with a lightweight prototype, consisting of a camera and an array of LEDs, as well as an off-the-shelf tablet. Our results are validated against reconstructions from a professional 3D scanner and photographs, and compare favorably with state-of-the-art techniques.

Index Terms—photometric stereo, illumination multiplexing, feature learning, neural acquisition

1 Introduction

Free-form scanning of 3D geometry in the presence of complex appearance is an important problem in computer graphics and computer vision. It is useful for various applications including e-commerce, visual effects, 3D printing, and cultural heritage.

Despite extensive research on traditional shape reconstruction over the past decades, this problem remains challenging. At one hand, multi-view stereo [1] usually assumes a Lambertian-dominant reflectance in computing reliable view-invariant features. Complex appearance variation with view or lighting is not welcome. It may alter the native spatial features on object surface, or specularly reflect the projected pattern from active illumination, either of which may result in correspondence matching errors. On the other hand, photometric stereo [2], [3] takes as input the images under varying illumination at the *same* view(s), with the help from additional hardware (i.e., a tripod) for fixing the view. In free-form scanning, however, such images are impractical to acquire, as the camera/view is *constantly changing*.

Recently, image-driven differentiable optimization makes a significant success in geometric reconstruction [4], [5], [6]. The geometry and appearance are optimized jointly, with a loss function that encourages the rendering results to approximate corresponding input images in an end-to-end fashion. But for complex appearance such as highly specular or strongly anisotropic reflectance, the result quality is not yet satisfactory, due to the insufficient physical sampling capability (e.g., a flash only produces a single point sample in the illumination domain at a time [7]) and the insufficient fidelity of appearance representation.

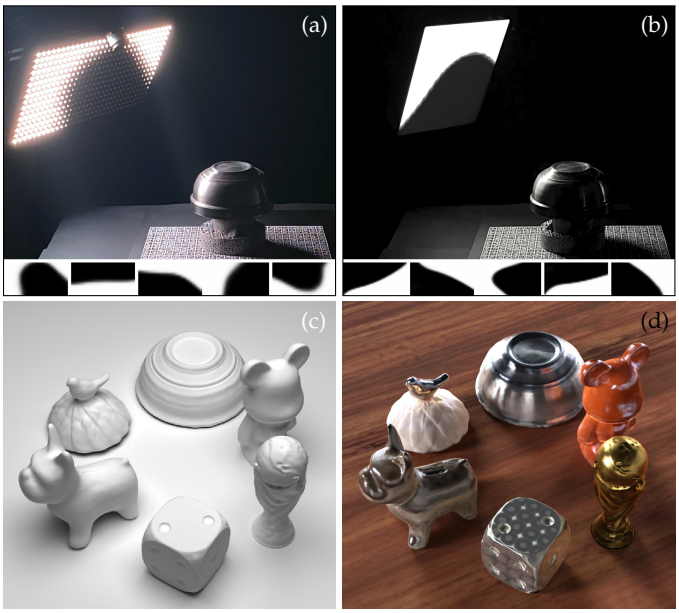

Fig. 1: Using an illumination-multiplexing device, such as a lightweight prototype consisting of a single camera and a programmable light array (a) or an off-the-shelf tablet (b), we propose a system that learns to acquire with pre-optimized time-varying lighting patterns (the bottom insets in (a) and (b)) at unstructured views, and reconstruct both the geometry (c) and complex anisotropic reflectance (d) of a number of challenging objects. Please refer to the supplementary video for animated rendering results.

To tackle the above difficulties, we present a novel framework to automatically learn to aggregate and transform photometric measurements from multiple *unstructured* views into spatially distinctive and view-invariant features (Fig. 2). This low-level transform is modular, and can

• *K. Zhou and H. Wu are the corresponding authors. All authors are with the State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310058, China, except that P. Tan is with Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China. K. Kang is additionally affiliated with the Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia.*
E-mail: {kunzhou,hwu}@acm.org

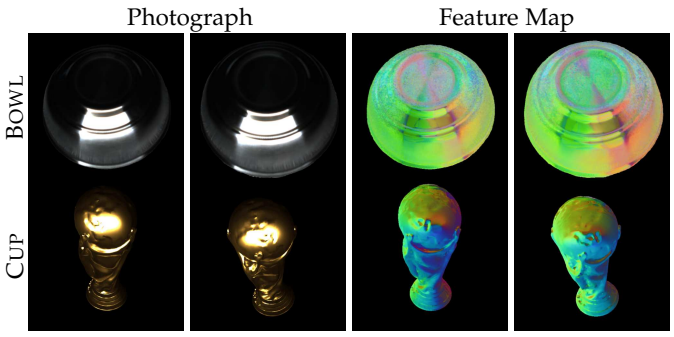

Fig. 2: Visualization of our feature maps at 2 views (3rd/4th column) on a scanned object (top row) and a synthetic one (bottom row). The 1st/2nd column are input photographs at the same view as the 3rd/4th column, respectively. Our high-dimensional features are projected to 3D via PCA for visualization.

enhance any multi-view stereo pipeline as a preprocessing step. To encode more angular information for high-quality reconstruction, we employ pre-optimized time-varying illumination multiplexing to physically convolve with a BRDF slice to produce a photometric measurement. To handle varying views, we carefully warp related measurements to preserve their geometric relationships for efficient neural processing. The illumination conditions during acquisition and the feature transform are jointly trained on a large amount of synthetic data. Our data-driven framework is highly flexible and can adapt to various factors, including the physical capabilities/characteristics of different setups, and different types of appearance. Since our photometric measurements reveal useful information about appearance as well, we further build a system to scan and reconstruct both the geometry and reflectance for complete object digitization.

The effectiveness of our system is demonstrated on scanning a number of challenging objects with a wide variation of shape and reflectance (Fig. 1). Our framework is not tied to any particular setup (e.g., a lightstage [8]). We conduct the experiments on a lightweight illumination-multiplexing prototype [9], consisting of a camera and an LED array, as well as an off-the-shelf tablet using its front camera and its screen as a programmable light source. Our shape results are validated against reconstructions from a professional 3D scanner, and our appearance results against photographs. We compare favorably with state-of-the-art techniques both in terms of geometry and reflectance.

2 Related Work

Due to the space limit, below we only review previous work that is closely related to our approach.

2.1 Photometric Stereo

These techniques compute a normal field from appearance variations under different lighting and typically at a fixed view, and then integrate into a depth map [3], [10]. Research efforts have been made from the original assumption of a Lambertian reflectance and a calibrated light, to more general appearance [11], [12], [13] and/or uncalibrated lighting conditions [14], [15], [16]. Due to indirect measurements, the integrated depth map often suffers from low-frequency distortions. Multi-view photometric stereo leverages photometric cues at multiple views to obtain a complete 3D shape. These methods refine an initial coarse geometry with the estimated normals [17], [18], [19]. Vlasic et al. [20] directly combine multi-view depth maps, each of which is computed from a normal field, to produce the final result. Logothetis et al. [21] exploit the relationship between a signed distance field and normals for reconstruction. Bi et al. [22] build a rig of 6 colocated cameras and lights, and estimate normals from sparse multi-view photometric images to refine an initial geometry. Yang et al. [23] jointly estimate geometry, materials and lighting with differentiable rendering. The closest work to ours is EPFT [24]. They learn to probe the angular information with a set of lighting patterns from a fixed view at a time, and transform the measurements to useful multi-view feature maps for 3D reconstruction.

All above work requires taking multiple photographs at a fixed view; none can be applied to free-form scan, in which the camera is constantly moving relative to the physical sample. In comparison, we propose a data structure and network to efficiently aggregate and transform photometric measurements at unstructured views, which are unique in free-form scanning, to low-level geometric features.

2.2 Multi-view Stereo

Traditional methods extract low-level features from each image, compute feature correspondences across multiple views, and apply triangulation to obtain 3D information [1]. Spatial aggregation is typically required, as the raw measurements at each pixel are not distinctive enough to establish reliable correspondences. While excellent results are achieved on Lambertian-dominant appearance [25], [26], the reconstruction quality cannot be guaranteed in the presence of complex materials that vary with view or lighting [27], [28]. Recently, hand-crafted features are replaced with automatically learned ones [29], [30], [31]. However, learning high-quality features in the presence of complex materials remains difficult, due to the lack of corresponding training datasets.

With the advances in machine learning, considerable progress has been made in developing end-to-end frameworks which take as input photographs and directly predict depth maps. Yao et al. [32] encode camera geometries in the network as differentiable homography, and construct 3D cost volumes to regress per-view depth map. Gu et al. [33] introduce cascade cost volume to predict per-view depth map in a coarse-to-fine manner. These approaches do not exploit physical appearance information for 3D reconstruction.

Our work is orthogonal to most techniques here, which focus on efficient processing in the spatial domain. In comparison, we learn how to aggregate useful information in the high-dimensional view-illumination domain for enhanced geometric reconstruction. The optimized lighting patterns essentially serve as convolution kernels to actively probe the *angular* domain. Unlike the majority of related work,

which tries to exclude photometric information, we exploit such information to efficiently handle highly challenging anisotropic appearance. Moreover, our learned low-level feature transform can be plugged in any existing multi-view stereo pipeline as a preprocessing module.

2.3 Image-driven Differentiable Optimization

Recent research optimizes both shape and appearance, using captured images as a form of self-supervision [4], [6], [34], [35], [36]. Various neural geometric representations are proposed, along with a shading network to account for appearance variations [4], [6], [35], [36]. While the network considerably improves the reconstruction robustness compared with the Lambertian model, it cannot model challenging appearance like anisotropic materials with high fidelity.

A number of techniques [22], [37], [38], [39] focus on recovering detailed geometry and materials using a point light colocated with the camera. These methods struggle with strong specular highlights or anisotropic reflections, due to the extremely low sampling efficiency in the angular domain. Another line of work optimizes geometry and appearance under environment lighting [7], [34], [40], [41], [42], [43]. Fundamentally, the material reconstruction of passive photometric approaches is limited by the frequency distribution of the environment illumination [44]. As a result, the quality of jointly optimized geometry is affected.

Our approach is different mainly from 3 aspects. First, we jointly optimize the illumination condition during acquisition to pack more useful information in the measurements for improved reconstruction. The light array we use also has a substantially higher physical sampling capability compared with a flash. Second, we leverage the existing domain knowledge on appearance (i.e., the GGX BRDF model) in training our network, while such knowledge is entirely learned from measurements on a per-object basis in the majority of related work. Finally, we do not jointly optimize the shape and appearance. This decoupling prevents appearance reconstruction errors from propagating to geometry results [5].

3 Acquisition Setup

We conduct acquisition experiments on two illumination-multiplexing devices, a lightweight custom-built scanner similar to [9] (Fig. 1-a) and an off-the-shelf tablet (Fig. 1-b). The intrinsic/extrinsic parameters of the camera, as well as the positions, orientations, angular intensity and spectral distribution of the light sources, are carefully calibrated. We acquire 30/10 images per second on the scanner/tablet during scanning, respectively.

Prototype Scanner. Our prototype consists of a rectangular RGB LED array and a single machine vision camera. The LED array has 32×16=512 RGB LEDs, with a pitch of 1cm and a maximum total power of 40W. The intensity of each LED is independently controlled, and quantized with 8 bits per channel for implementation via Power Width Modulation (PWM). A 5MP Basler acA2440-75uc camera is mounted on the top edge of the LED array. A house-made circuit board is in charge of high-precision synchronization between the camera and the LED array.

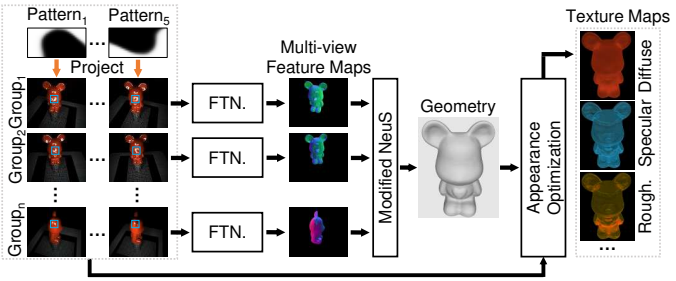

Fig. 3: Our runtime pipeline. First, we partition continuously captured images into groups of 5, each acquired under a different lighting pattern. Next, we crop patches from each image in the group, centered at a same pixel location. A network (Feature Transform Network) then transforms these data into a per-pixel high-dimensional feature at that location, the collection of which forms a feature map for the center view. We feed the feature maps from every group into a multi-view stereo pipeline for 3D reconstruction. With the computed shape, the appearance of the object is differentiably optimized with respect to all input images, and then stored as texture maps of GGX BRDF parameters. FTN. = Feature Transform Network.

Tablet. Our tablet is a 12.9-inch iPad Pro (4th gen.). We use its screen as a programmable light source, and employ its front-facing 7MP camera to take photographs. Note that the power of the iPad screen is considerably lower than our scanner, which translates to a higher requirement on handheld stability to avoid blurred images.

4 Preliminaries

The following equation describes the relationship among the image measurement B from a surface point $\mathbf{p}$, the reflectance f and the intensity I of each LED of the scanner/each pixel of the tablet, for a single channel.

$$B(I;\mathbf{p}) = \sum_l I(l) \int \frac{1}{||\mathbf{x_l}-\mathbf{x_p}||^2} \Psi(\mathbf{x_l}, -\omega_\mathbf{i}) V(\mathbf{x_l}, \mathbf{x_p}) f(\omega_\mathbf{i}; \omega_\mathbf{o}, \mathbf{p})(\omega_\mathbf{i}\cdot\mathbf{n_p})^+(-\omega_\mathbf{i}\cdot\mathbf{n_l})^+ d\mathbf{x_l}. \quad (1)$$

Here l is the index of a locally planar light source, and $I(l)$ is its intensity in the range of [0, 1], the collection of which with each possible l is a **lighting pattern**. Moreover, $\mathbf{x_p}/\mathbf{n_p}$ is the position/normal of $\mathbf{p}$, while $\mathbf{x_l}/\mathbf{n_l}$ is the position/normal of a point on the light with an index of l. We denote $\omega_\mathbf{i}/\omega_\mathbf{o}$ as the lighting/view direction. $\Psi(\mathbf{x_l}, \cdot)$ represents the angular distribution of the light intensity. V is a binary visibility function between $\mathbf{x_l}$ and $\mathbf{x_p}$. The operator $(\cdot)^+$ computes the dot product between two vectors, and clamps a negative result to zero. Finally, f is a 2D slice of anisotropic GGX BRDF [45].

As B is linear with respect to I (Eq. 1), it can be expressed as the dot product between I and a **lumitexel** c:

$$B(I;\mathbf{p}) = \sum_l I(l) c(l;\mathbf{p}),$$
$$c(l;\mathbf{p}) = B(\{I(l)=1, \forall_{j\neq l} I(j)=0\};\mathbf{p}), \quad (2)$$

where c is a function of the light index l, defined on the surface point p of the sampled object [46].

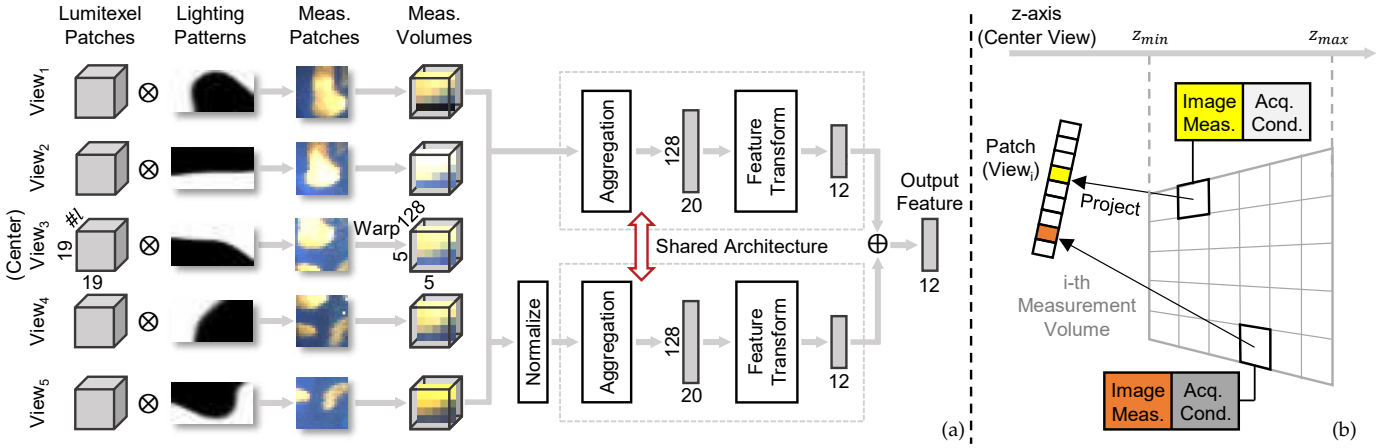

Fig. 4: Our network (a) and warping illustration (b). The network (a) takes as input the lumitexels corresponding to all pixels in 5 patches from a group, and encodes them as measurements by simulating lighting pattern projections. The measurements of each view are warped to their respective measurement volume (b), defined at the same center view. A total of 5 volumes are aggregated and transformed to a feature vector, by combining the outputs from an unnormalized and a normalized network branch which share the same architecture. A 2D warping example of a patch at the i-th view is shown in (b). First, a measurement volume is set up with respect to the center view. We then fill each voxel by projecting its center to the patch, and fetching the corresponding image measurement. The acquisition condition, including view and lighting information, is also stored. Meas. = measurement, Acq. = Acquisition, Cond. = condition.

5 Overview

To scan an object, we move an illumination-multiplexing device around it to take photographs continuously. The light source on the device is programmed to loop over $\#p$ pre-optimized lighting patterns. The camera exposure is synchronized with the pattern projection. We define every $\#p$ consecutively captured images (i.e., views) as a **group**, with the first image acquired with the first lighting pattern, etc.

To reconstruct the geometry (Sec. 6), for each group, we aggregate and transform the multi-view photometric information from all $\#p$ images into a high-dimensional feature map, defined at the **center view** (i.e., the view of the $\lceil \#p/2 \rceil$-th image). The feature maps of all groups are directly sent as input to a multi-view stereo pipeline (e.g., NeuS [4]) for 3D reconstruction. To recover the reflectance (Sec. 7), we optimize the GGX BRDF parameters with respect to input photographs, given the previously reconstructed mesh. The results are stored as texture maps, which can be rendered with any standard pipeline under novel view and illumination conditions. Please see Fig. 3 for an illustration.

Note that each group is actually a small set of frames similar in view, but different in lighting. This structure generalizes from standard photometric stereo, whose input images are taken with varying illumination at the *same* view. In comparison, the images in one of our groups have slightly *different* views, as it is impossible to maintain a fixed view with a handheld device during acquisition (i.e., one key challenge we address in this paper).

6 Feature Transform Network

It models physical acquisition and computational reconstruction together, to enable the automatic, joint optimization of both processes (Fig. 4). For training, the network input are $\#p$ patches of lumitexels, representing the spatially-and-angularly varying appearance at $\#p$ views in a group. These patches of lumitexels are encoded by corresponding lighting patterns to simulate the measurement process (Sec. 6.1). Next, to handle unstructured views and unknown depths, each patch of measurements are warped to a measurement volume defined at the center view (Sec. 6.2). Finally, all warped measurements are aggregated (Sec. 6.3), and transformed to a high-dimensional feature vector, corresponding to the center pixel of the patch at the center view (Sec. 6.4). At runtime, for each group, we slide a window over the image domain, crop from all captured images in the group, send the resulting $\#p$ patches of measurements to our network, and assemble the final *per-pixel* feature vectors as a feature map. Fig. 5 visualizes our network architecture. We use $\#p$=5 in all experiments.

6.1 Encoding

The first part of our network is an encoder that maps the measurement process. It consists of a linear fully-connected (fc) layer, whose weights correspond to the lighting patterns in acquisition. The output are 5 patches of 19×19 pixels, corresponding to the 5 views in a group. Essentially, each output pixel represents an image measurement under a lighting pattern from a particular view, modeled as the dot product between an input lumitexel and a pattern (Eq. 2).

Note that increasing the patch size would increase the computation cost, while decreasing it would reduce the chance of capturing essential information for transforming to a high-quality feature. The current patch size is determined via experiments.

6.2 Warping

To represent the potential geometric relationships between measurements, the second part of our network warps each

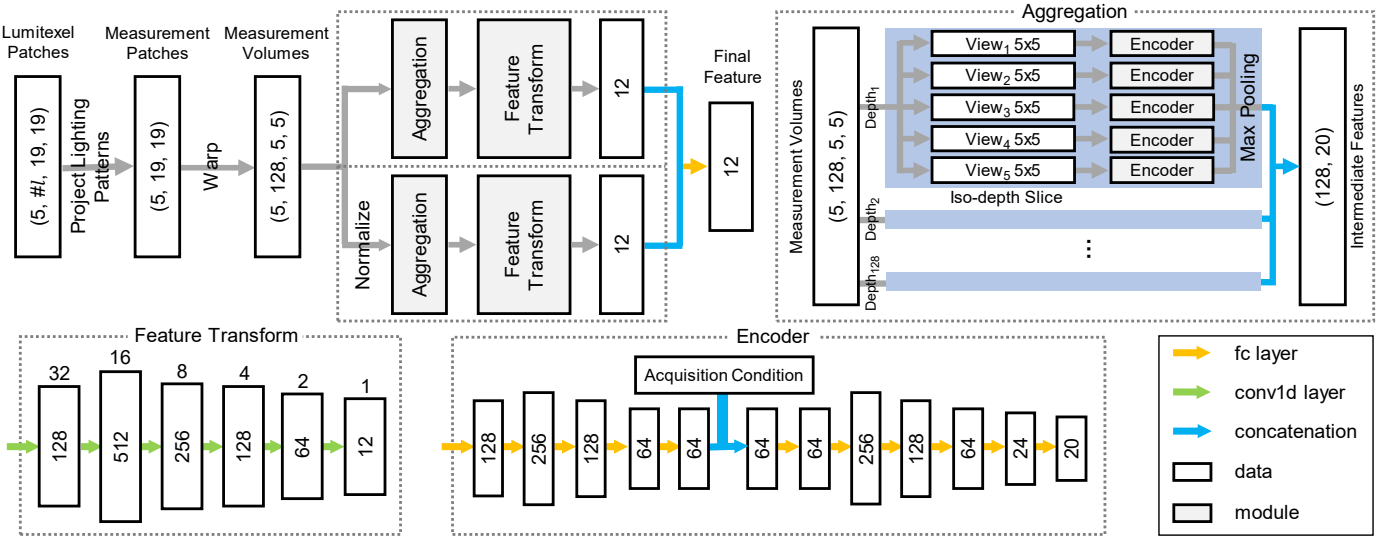

Fig. 5: Network architecture. Our network takes as input the lumitexels corresponding to all pixels in 5 patches from a group, and encodes them as measurements by simulating lighting pattern projections. The measurements of each view are warped to a measurement volume. The total of 5 volumes are aggregated and transformed to a feature vector, by combining the outputs from an unnormalized and a normalized branch that shares the same structure. The dimension of data is specified in the corresponding block. In the feature transform module, the dimension of depth is additionally specified on the top of each block. In the aggregation module, we loop over each of 5 iso-depth slices within 5 measurement volumes and transform them into a 20D intermediate feature. In the end, all intermediate features at 128 different depth hypotheses are aggregated to a final 12D feature.

patch to a **measurement volume** (Fig. 4-b). As a result, 5 patches in a group are warpped to 5 different measurement volumes. This step is critical for efficient handling of free-form scanned data with unknown depths.

First, we define a measurement volume as a view frustum at the *same* center view with a resolution of $5 \times 5 \times 128$. Specifically, we cast rays from the camera center towards each pixel in the center 5×5 window of the patch at the center view. Each ray is discretized into 128 depth hypotheses, uniformly sampled in the range of $[z_{\min}, z_{\max}]$. The depth range $z_{\min}/z_{\max}$ is calculated from a coarse bounding box (or can be manually specified). Ideally, a volume of $1 \times 1 \times 128$ is sufficient to model *depth uncertainty* for the center pixel of a patch. However, due to inevitable registration errors in practice, we enlarge the lateral neighborhood to 5×5 to tolerate such inaccuracies.

Next, we convert each patch into its own measurement volume (Fig. 4-b). Specifically, we loop over all voxels in the volume, project each voxel center to the patch, and store the corresponding measurement (black if the projection falls outside the patch) along with the acquisition condition to the current voxel. Our acquisition condition includes (1) one hot-encoding of the lighting pattern index, (2) the center pixel location of the current patch in the image, (3) the depth of the current voxel, (4) the camera transform relative to the center view, and (5) the camera pose of the center view. The above information is stored for each voxel, as we want our network to be aware of the factors that are related to final geometric features. Note that the idea of encoding acquisition conditions for neural processing is proposed in [9] for free-form appearance scanning.

In the end, a voxel in the i-th measurement volume contains the image measurements from the i-th view, whose corresponding 3D positions *might* fall within the voxel.

6.3 Aggregation

The third part of the network aggregates the information from all 5 measurement volumes, each of which is computed from a patch. First, for each iso-depth slice in one volume, we flatten and transform it to a lower dimensional latent vector using an encoder. As a result, each volume is converted to 128 latent vectors. This step can be viewed as an aggregation along the lateral dimensions. Next, for each depth, we aggregate across 5 views by performing max pooling on all related latent vectors, and store the result as an intermediate feature. Note that this step aggregates illumination as well, since each view is associated with a different lighting pattern. The output of this part is 128 intermediate features. Now the only dimension that has not been aggregated is the depth, which is left for the next step.

6.4 Feature Transform

This fourth part produces the final per-pixel geometric feature, the collection of which will be fed to multi-view stereo for 3D reconstruction. We send all 128 intermediate features to a convolutional neural network to extract a 12D feature as output. This allows the network to automatically learn how to "softly" select the most matching depth (among all 5 views) as well as its corresponding feature.

The above architecture (Sec. 6.3-6.4) is repeated to build two branches, as illustrated in Fig. 4-a & 5. One unnormalized branch works exactly as described above, while the other branch will normalize the volume across multiple views on a per-voxel basis. The idea is to prevent the

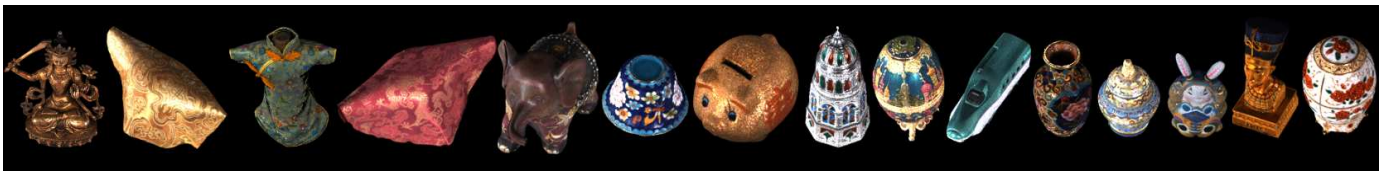

Fig. 6: Our training dataset of 15 high-quality objects, digitized by a commercial 3D scanner and a professional light stage [48].

network from learning stable diffuse albedos only. After normalization, the absolute values of albedos no longer matter, therefore forcing the network to exploit other useful sources of information. Finally, the two 12D output vectors from both branches are combined into a final 12D feature via a linear fc layer, similar to [24].

6.5 Loss Function

The function is defined as follows:

$$L = \lambda_0 L_0 + \lambda_1 L_1 + \lambda_2 L_2 + \lambda_p L_p, \tag{3}$$

where L_0, L_1, L_2 are the similarity loss [47] on the final feature, and the feature from the unnormalized/normalized branch, respectively. The similarity loss is defined as:

$$L_{\text{sim}} = \frac{1}{2}(\sum_i \log s^c_{ii} + \sum_i \log s^r_{ii}),$$
$$s^c_{ij} = \frac{\exp(-d_{ij})}{\sum_m \exp(-d_{mj})}, s^r_{ij} = \frac{\exp(-d_{ij})}{\sum_n \exp(-d_{in})}.$$

Here d_{ij} is the Euclidean distance matrix of features in a batch of training samples. In each batch, we consider the features of the *same* 3D point at different views, as well as the features corresponding to *different* points. The e1 loss essentially decreases the distances of the former (view invariance), while increases the distances of the latter (spatial distinctiveness). Finally, L_p is a loss term on the brightness of lighting patterns, as brighter patterns are desired for high SNR acquisition during scanning, defined as: $L_p = \sqrt{\#l / \sum_l I(l)}$, where $\#l$ is the number of independently controlled light sources. We use $\lambda_0 = 0.33, \lambda_1 = 0.33, \lambda_2 = 0.33$ and $\lambda_p = 0.05$ in our experiments.

6.6 Training

We implement our network with PyTorch, using the Adam optimizer with batch size of 32 and a momentum of 0.9. Xavier initialization is applied to all weights in the network. We train 100K iterations with a learning rate of 10^{-4}, which takes 8 hours to finish.

The training data are synthetically generated using 15 pre-captured objects (Fig. 6) and captured scanning motions. Each object consists of a 3D mesh along with texture maps of GGX BRDF parameters. 25 scanning processes are recorded, each of which consists of 2,000 consecutive camera poses. We compute the relative motion of every 5 views with respect to the center view, which will be processed to synthesize the camera motion for training a single group. Note that our approach is *not tied* to the current data since it is entirely data-driven.

Specifically, the synthetic data for a group are generated as follows. We first synthesize the center view. To do so, we randomly sample a 3D point on a random object (Fig. 6) and a visible view direction based on its geometry normal. For the viewing direction, a camera position is randomly generated along the direction within a predefined range of valid distance (16-65cm in our experiments). Next, a random foreground pixel is selected as the projected position of the 3D point. With the projected pixel and the position of the 3D point, a camera pose is initialized, and then we randomly rotate it around the view direction. To produce other views in the group, we random select part of pre-captured continuous scanning motions, and use the relative tranforms between frames to synthesize all other 4 views. Finally, we compute the lumitexels for all pixels in the patch of each view, whose center is the projected pixel, by ray-tracing to the object surface and simulating the light reflections with the associated BRDF parameters, according to (2). The patches of lumitexels will be used to synthesize images measurements (Sec. 6.1). Please also refer to [24] for descriptions on a similar process.

7 Appearance Optimization

After geometric reconstruction, we establish a uv-parameterization over object surfaces, and compute BRDF parameters at each valid texel via differentiable optimization. For a specific texel, we first project the corresponding 3D position to all visible views to gather its image measurements under learned lighting patterns. We reparameterize the GGX BRDF model plus the local frame with a 16D latent code and jointly train a fully-connected network that transforms the latent code to GGX BRDF parameters and the local frame as in [49], by minimizing the difference between rendering results (Eq. 1) and the gathered measurements. Finally, we convert the latent code at each texel to anisotropic GGX BRDF parameters and store them in texture maps as the appearance result. Please refer to the supplementary material for more details on appearance modeling.

8 Implementation Details

We remove over-blurry images from our sequence to avoid the negative impact over the final results. We calculate the level of blurriness for each image [50], and discard an entire group if the blurriness of any image in the group reaches a threshold. For each remaining image, we perform structure-from-motion with COLMAP [51] to compute camera poses from the ARTags [52] placed along with the object (Fig. 1-a & b). We apply SAM [53] to segment the object from the background for each center-view image. After geometry

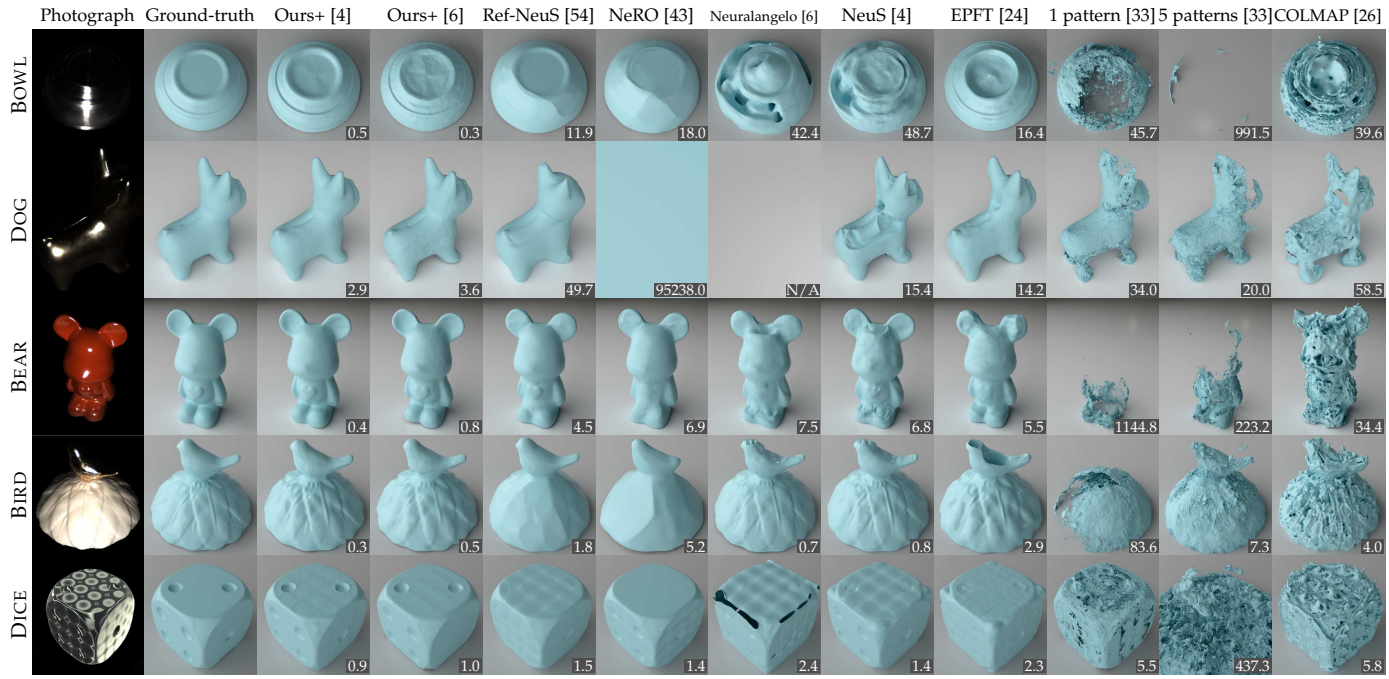

Fig. 7: Comparison with different geometric reconstruction techniques. From the left to right, a photograph, ground-truth geometry, our results with NeuS [4]/Neuralangelo [6] as the backend, Ref-NeuS [54], NeRO [43], Neuralangelo [6], NeuS [4], EPFT [24], CasMvsNet [33] with 5/1 lighting pattern(s), and COLMAP [26]. Vanilla Neuralangelo, NeuS and COLMAP take input photographs captured under an indoor office environment lighting. EPFT is re-trained with a single learnable lighting pattern to adapt to free-form scanning. CasMVSNet is supplied with input photographs under 1(corresponding to our center view) or 5 of our learned patterns. BOWL, DOG, BEAR, BIRD are real objects, and DICE is synthetic. Note that Neuralangelo fails to produce a mesh for DOG. Quantitative errors in Chamfer distance are reported in the bottom right corner of related images.

reconstruction, a uv-parameterization with a texture resolution of 1024×1024 is generated for appearance optimization (Sec. 7).

In terms of lighting patterns for the tablet, we use a resolution of 27×36, which is much lower than the native one, to save the otherwise prohibitively expensive computational costs. We modify the original NeuS [4] by changing the output dimension of the last fully-connected layer to 12, the same dimension as our feature vector. Similar modification is applied to Neuralangelo [6], which changes its output dimension to 12. Please refer to the supplementary material for more details.

9 RESULTS & DISCUSSIONS

We use our prototype to scan 4 real objects with complex appearance (BOWL, DOG, BEAR and BIRD). The maximum dimension of each object ranges from 6 to 18 cm. The reconstructed geometry and appearance are shown in Fig. 7 & 8. The iPad is used to scan BOWL (Fig. 9). It takes about 60 seconds to scan a real object. The remaining objects (DICE, CUP, NAJADE, and MATBALL) are synthetic, whose images are computed via physically based rendering with a virtual scanner. The SVBRDF of MATBALL is SATIN0112 from [55].

All computation is performed on a server with dual AMD EPYC 7763 CPUs, 768GB DDR4 memory and 8 NVIDIA GeForce RTX 4090 GPUs. All results are rendered with NVIDIA OptiX. It takes 70 minutes to preprocess the data of an object, including blurriness computation, object segmentation and camera pose estimation. For a group of 5 images, our network needs 2.5 seconds to transform them to a feature map. On average we use 60 groups to reconstruct an object. The geometry reconstruction via NeuS/Neuralangelo takes 6/14 hours respectively, and the appearance optimization about 1 hour. Samples of the captured images under our optimized lighting patterns and corresponding feature maps are visualized in Fig. 2.

9.1 Comparisons

In the following comparisons, our approach uses on average 60 groups of images (#images = 300) for reconstructing each object, while competing methods use around 300 images for fairness. In appearance comparisons, we randomly choose 70% of the captured images for training and use the remaining 30% as test images to evaluate novel view and lighting synthesis. The ground-truth geometry of a physical object is obtained with a commercial 3D scanner [56]. Due to the challenging appearance (e.g., strong anisotropic/specular reflections), we have to apply fine powder to object surfaces for the scanner to work properly.

Geometry Reconstruction. In Fig. 7, we compare our geometry reconstruction results with state-of-the-art methods on 5 objects with challenging appearance, including

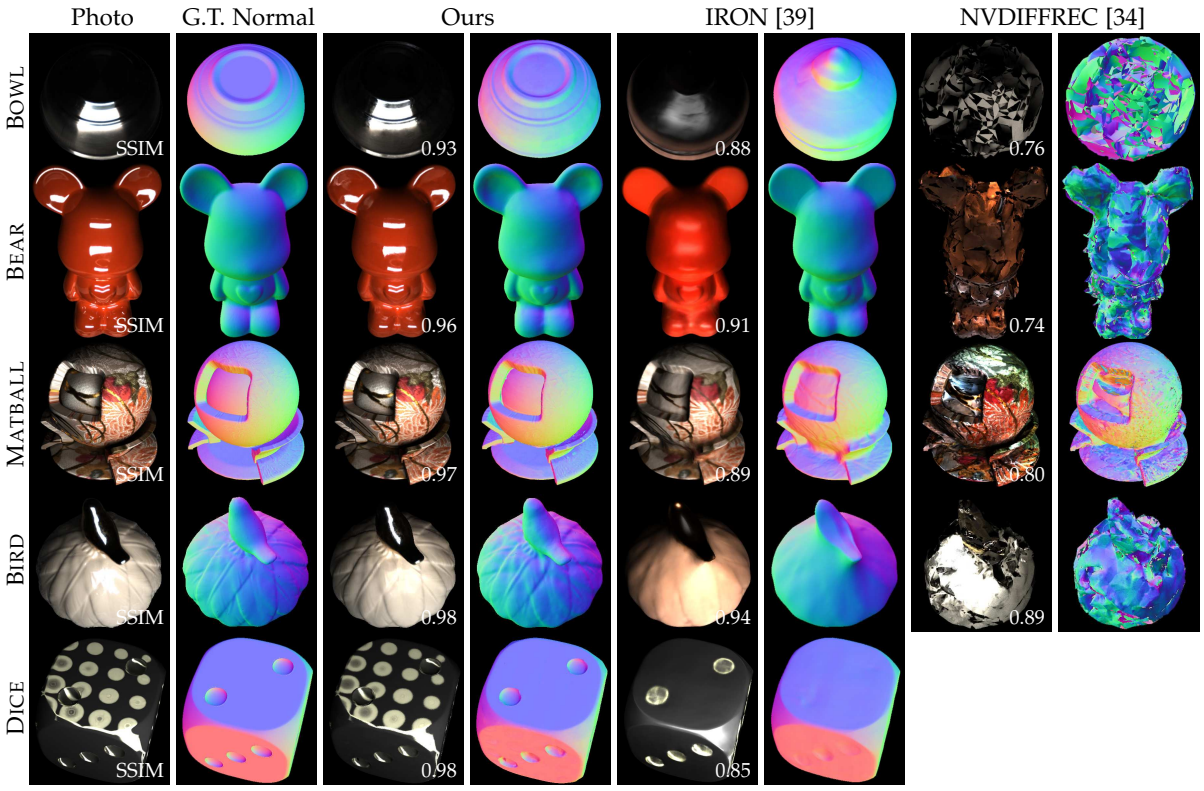

Fig. 8: Comparison with techniques on joint optimization of shape and appearance. For each pair of images, the left is a photograph/appearance rendered with novel view and lighting, and the right a normal map from the corresponding geometry. From the left to right: ground-truth, our reconstructions, the results from IRON [39], and NVDIFFREC [34]. Note that NVDIFFREC fails to reconstruct DICE due to strong anisotropic appearance. The input images are captured with a point light for IRON, and with an indoor office environment lighting for NVDIFFREC. Quantitative errors in SSIM are reported in the bottom right corner of related images.

Fig. 9: Comparison of reconstructions of the same object using our prototype and tablet. The main image shows the shape, while the inset is the rendered appearance with novel view and lighting.

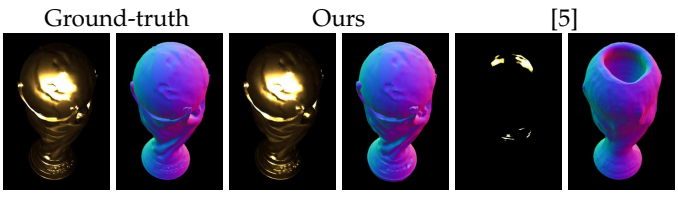

Fig. 10: Comparison between our reconstruction and [5]. For each pair of images, the left one is the appearance rendering, and the right a normal map from the corresponding geometry.

Fig. 11: Appearance reconstruction rendered with novel environment lighting. The top row are our relighting results, and the bottom row results from NeRFactor [7], which fails to reconstruct the strong anisotropic reflectance of DICE.

specular reflections and textureless regions. The BOWL and DICE also exhibit strong anisotropic appearance. For a fair comparison, the acquisition lighting condition for Ref-NeuS, NeRO, Neuralangelo, NeuS and COLMAP is a conventional indoor office environment. This is because our pilot study shows that for these methods, using input photographs under our lighting patterns leads to lower reconstruction quality. For CasMVSNet, we test with input photographs under 1(center-view) or 5 of our learned patterns. Due to the lack of efficient handling of complex appearance variations, Fig. 7 shows unsatisfactory reconstructions from Ref-Neus, NeRO, Neuralangelo, NeuS, CasMVSNet or COLMAP.

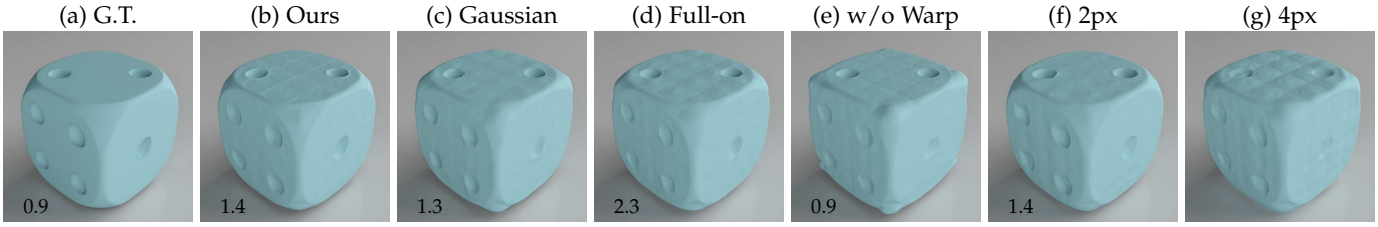

Fig. 12: Impact of lighting patterns, warping and camera pose errors over geometric reconstruction. From the left to right: the ground-truth, the reconstruction using our network, ours with 5 Gaussian noise lighting patterns, ours with a full-on pattern, ours without the warping step and ours with perturbed camera poses (average reprojection error = 2px/4px). Quantitative errors in Chamfer distance are reported in the bottom right corner.

TABLE 1: Correlation between transformed features and various parameters, averaged over our synthetic training dataset. From the 2nd column to the 4th, our features from unnormalized/normalized branch, and the final combined features (c.f. Sec. 6.4). Higher values indicate stronger correlations.

Correlation	Unnormalized Branch	Normalized Branch	Combined Result
Diffuse	0.89	0.42	0.89
Specular	0.68	0.38	0.68
Roughness	0.49	0.41	0.52
Normal	0.55	0.95	0.94
Tangent	0.39	0.62	0.62
Depth	0.46	0.54	0.56
Position	0.58	0.83	0.84

Moreover, the effectiveness of EPFT [24] is limited, as it relies on photometric information captured from a fixed view. Our results (the last two columns of Fig. 7) outperform other methods both qualitatively and quantitatively.

Joint Reconstruction of Geometry & Appearance. In Fig. 8, we compare both geometry and appearance reconstructions with related methods. For a fair comparison, IRON [39] takes as input photographs with a co-located flash, and the acquisition lighting for NVDIFFREC [34] is the same indoor office environment as used in Fig. 7, similar to the original papers. We also find input photographing under our patterns results in lower-quality reconstructions with their methods. Our approach outperforms competing methods qualitatively and quantitively, as we can handle challenging appearance by efficiently probing the angular domain with learned illumination multiplexing.

In Fig. 10, we compare with a state-of-the-art differentiable optimization technique [5]. Their method struggles to accurately reconstruct the geometry in the presence highly specular reflectance. The end-to-end, joint optimization for shape and appearance is challenging to converge to correct results. Plus, their point light does not have sufficient sampling capability, leading to an under-constrained optimization. We also compare with NeRFactor [7] on appearance reconstruction in Fig. 11, by rendering the results under novel view and light. Their approach relies on precise estimation of the density field, which becomes inaccurate in the presence of complex appearance. This leads to unsatisfactory reconstructions, and consequently low-quality renderings.

9.2 Evaluations

We first analyze the correlations between our learned features with various common parameters in Tab. 1, by computing Canonical Correlation Analysis (CCA) between our features and individual parameters, averaged over our synthetic training dataset.

In Fig. 12, we evaluate the impact of lighting patterns, warping and camera pose errors on reconstructing the geometry of DICE. We first compute the shape with our approach on images rendered with camera motions different from training, as shown in (b). Next, we evaluate the impact of different lighting patterns. (c)/(d) shows the result computed using our network trained with 5 fixed Gaussian noise patterns/a full-on pattern. The effectiveness of our learned patterns is clear, by comparing (b),(c) & (d). Moreover, we train a network without warping, which results in considerably higher reconstruction error, as shown in (e). This demonstrates the effectiveness of our warping, which efficiently models the 3D uncertainty. The last 2 images evaluate the robustness of our approach against errors in camera poses. We perturb the camera poses with Gaussian noise of different standard deviations as in [9], and report the average reprojection errors on top of the 2 images. Our approach can tolerate a reprojection error of 2 pixels, as shown in (f). However, when the error increases to 4, the 3D reconstruction quality degrades (g). For reference, the average reprojection error with standard SfM [51] is 0.6 pixel.

In Fig. 13, we evaluate the impact of the number of lighting patterns, two-branch architecture and scanning speed on reconstructing the geometry of NAJADE. Its appearance is textureless and highly specular (as shown in the first inset of the figure). We first compute the shape with our network trained on 3, 5, and 7 lighting patterns, as shown in Fig. 13 (b), (c) & (d). While using 5 patterns can improve geometric details over 3 patterns, adopting 7 patterns brings marginal benefits. Next, we assess the impact of normalized/unnormalized branches in Fig. 13 (e)/(f). For the textureluess NAJADE, normalized features clearly outputperform unnormalized ones. Finally, we evaluate the impact of scanning speed in Fig. 13 (g) & (h). Higher speed results in lower reconstruction quality, as the content in a group of patches becomes less correlated. This makes it more difficult to extract useful information from the input.

We further demonstrate the modular property of our features, by applying them to boost 3D reconstruction with

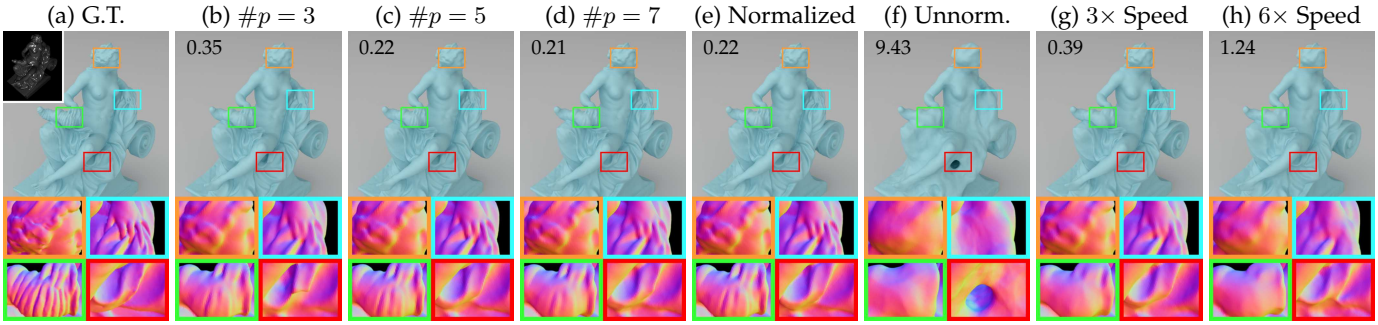

Fig. 13: Impact of the number of lighting patterns, two-branch architecture and scanning speed over geometric reconstruction. From the left to right: the ground-truth, reconstructions using our network trained with 3, 5, and 7 lighting patterns; reconstruction using features from normalized/unnormalized branch of our network; and reconstructions from $3/6\times$ the average scanning speed of the training dataset. The bottom insets visualize the reconstructed normals. Quantitative errors in Chamfer distance are reported in the top left corner. Unnorm. = Unnormalized.

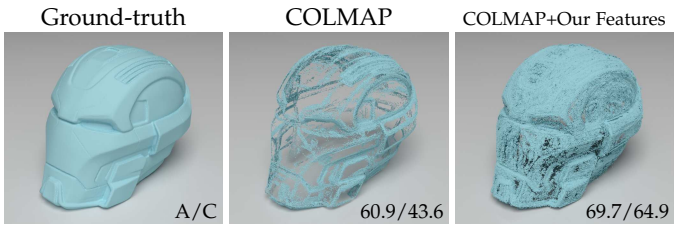

Fig. 14: Boosting another MVS method (COLMAP) with our features. From the left to right: the ground-truth, geometric reconstruction by COLMAP from environment-lit images, and the result by sending our feature maps to COLMAP. Quantitative errors in accuracy/completeness percentage are indicated in the bottom right corner.

a different backend, COLMAP. We test on a synthetic object with a homogeneous shiny material. In Fig. 14, we employ COLMAP to reconstruct a 3D shape from images of the object rendered under environment lighting, as well as feature maps computed with our network from the same number of images under learned lighting patterns. Considerable quality improvement is shown with the help of our learned features. Finally, we test the repeatability of our approach. Two students, who are not involved in this project, are asked to independently capture about the same number of photographs of the same object with our prototype. The reconstructed shapes, shown in Fig. 15, are visually similar.

Fig. 15: Repeatability experiment on our geometric reconstruction from 2 scans by 2 different students. Quantitative errors in Chamfer distance are reported in the bottom right corner.

10 Limitations & Future Work

Our work is subject to a number of limitations. First, our current imaging pipeline does not account for global illumination effects like inter-reflections. Also our system cannot capture transparent/translucent objects, which require special processing. In addition, we need a dark room for high signal-to-noise ratio acquisition, as the uncontrolled environment illumination is not modeled.

It will be interesting future work to address the above limitations. We are also interested in combing our system with a differentiable appearance scanner [9], to optimize lighting patterns for the acquisition of both shape and reflectance. It is also promising to combine our acquisition framework with an efficient and expressive neural representation [57]. Finally, we expect that the reconstruction quality could be further improved, if future tablets could offer APIs that enable hardware camera-screen synchronization.

Acknowledgments

The authors would like to thank Chong Zeng, Xiaohe Ma and Yizhong Zhang for their generous help and support. This work is partially supported by NSF China (62332015, 62227806 & 62421003), the Fundamental Research Funds for the Central Universities (226-2023-00145), the XPLORER PRIZE and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

References

[1] Y. Furukawa, C. Hernández *et al.*, "Multi-view stereo: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015.
[2] G. Chen, K. Han, and K.-Y. K. Wong, "Ps-fcn: A flexible learning framework for photometric stereo," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–18.
[3] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical engineering*, vol. 19, no. 1, p. 191139, 1980.
[4] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *NeurIPS*, 2021.
[5] C. Zeng, G. Chen, Y. Dong, P. Peers, H. Wu, and X. Tong, "Relighting neural radiance fields with shadow and highlight hints," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

[6] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-fidelity neural surface reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–18, 2021.

[8] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec, "Performance relighting and reflectance transformation with time-multiplexed illumination," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 756–764, 2005.

[9] X. Ma, K. Kang, R. Zhu, H. Wu, and K. Zhou, "Free-form scanning of non-planar appearance with neural trace photography," *ACM Trans. Graph.*, vol. 40, no. 4, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459679

[10] B. Shi, Z. Wu, Z. Mo, D. Duan, S.-K. Yeung, and P. Tan, "A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo," in *CVPR*, 2016, pp. 3707–3716.

[11] N. Alldrin, T. Zickler, and D. Kriegman, "Photometric stereo with non-parametric and spatially-varying reflectance," in *CVPR*, 2008, pp. 1–8.

[12] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz, "Shape and spatially-varying brdfs from photometric stereo," *TPAMI*, vol. 32, no. 6, pp. 1060–1071, 2009.

[13] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi, "Elevation angle from reflectance monotonicity: Photometric stereo for general isotropic reflectances," in *ECCV*. Springer, 2012, pp. 455–468.

[14] N. G. Alldrin, S. P. Mallick, and D. J. Kriegman, "Resolving the generalized bas-relief ambiguity by entropy minimization," in *CVPR*, 2007, pp. 1–7.

[15] R. Basri, D. Jacobs, and I. Kemelmacher, "Photometric stereo with general, unknown lighting," *IJCV*, vol. 72, no. 3, pp. 239–257, 2007.

[16] F. Lu, Y. Matsushita, I. Sato, T. Okabe, and Y. Sato, "Uncalibrated photometric stereo for unknown isotropic reflectances," in *CVPR*, 2013, pp. 1490–1497.

[17] C. Hernandez, G. Vogiatzis, and R. Cipolla, "Multiview photometric stereo," *TPAMI*, vol. 30, no. 3, pp. 548–554, 2008.

[18] M. Li, Z. Zhou, Z. Wu, B. Shi, C. Diao, and P. Tan, "Multi-view photometric stereo: a robust solution and benchmark dataset for spatially varying isotropic materials," *IEEE Transactions on Image Processing*, vol. 29, pp. 4159–4173, 2020.

[19] Z. Zhou, Z. Wu, and P. Tan, "Multi-view photometric stereo with spatially varying isotropic materials," in *CVPR*, 2013, pp. 1482–1489.

[20] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik, "Dynamic shape capture using multi-view photometric stereo," in *ACM SIGGRAPH Asia 2009 Papers*, 2009, pp. 1–11.

[21] F. Logothetis, R. Mecca, and R. Cipolla, "A differential volumetric approach to multi-view photometric stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1052–1061.

[22] S. Bi, Z. Xu, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi, "Deep reflectance volumes: Relightable reconstructions from multi-view photometric images," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 294–311.

[23] W. Yang, G. Chen, C. Chen, Z. Chen, and K.-Y. K. Wong, "Ps-nerf: Neural inverse rendering for multi-view photometric stereo," in *European Conference on Computer Vision*. Springer, 2022, pp. 266–284.

[24] K. Kang, C. Xie, R. Zhu, X. Ma, P. Tan, H. Wu, and K. Zhou, "Learning efficient photometric feature transform for multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 5956–5965.

[25] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *ICCV*, 2015, pp. 873–881.

[26] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV*, 2016.

[27] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg *et al.*, "The digital michelangelo project: 3d scanning of large statues," in *Proc. SIGGRAPH*, 2000, pp. 131–144.

[28] J. Salvi, J. Pages, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern recognition*, vol. 37, no. 4, pp. 827–849, 2004.

[29] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *TPAMI*, vol. 36, no. 8, pp. 1573–1585, 2014.

[30] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *CVPR*, 2015, pp. 4353–4361.

[31] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5695–5703.

[32] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 767–783.

[33] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2495–2504.

[34] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler, "Extracting triangular 3d models, materials, and lighting from images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8280–8290.

[35] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2492–2502, 2020.

[36] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[37] F. Luan, S. Zhao, K. Bala, and Z. Dong, "Unified shape and svbrdf recovery using differentiable monte carlo rendering," in *Computer Graphics Forum*, vol. 40, no. 4. Wiley Online Library, 2021, pp. 101–113.

[38] G. Nam, J. H. Lee, D. Gutierrez, and M. H. Kim, "Practical svbrdf acquisition of 3d objects with unstructured flash photography," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.

[39] K. Zhang, F. Luan, Z. Li, and N. Snavely, "Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[40] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. 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*, 2021, pp. 7495–7504.

[41] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. 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*, 2021, pp. 5453–5462.

[42] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 684–12 694.

[43] Y. Liu, P. Wang, C. Lin, X. Long, J. Wang, L. Liu, T. Komura, and W. Wang, "Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images," in *SIGGRAPH*, 2023.

[44] H. Wu, Z. Wang, and K. Zhou, "Simultaneous localization and appearance estimation with a consumer rgb-d camera," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 8, pp. 2012–2023, 2015.

[45] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet Models for Refraction through Rough Surfaces," in *Rendering Techniques (Proc. EGWR)*, 2007.

[46] H. P. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-based reconstruction of spatial appearance and geometric detail," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 2, pp. 234–257, 2003.

[47] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 661–669.

[48] K. Kang, C. Xie, C. He, M. Yi, M. Gu, Z. Chen, K. Zhou, and H. Wu, "Learning efficient illumination multiplexing for joint capture of reflectance and shape," *ACM Trans. Graph.*,

vol. 38, no. 6, pp. 165:1–165:12, Nov. 2019. [Online]. Available: http://doi.acm.org/10.1145/3355089.3356492
[49] X. X. Xu, L. Yuxin, H. Zhou, C. Zeng, Y. Yu, K. Zhou, and H. Wu, "A unified spatial-angular structured light for single-view acquisition of shape and reflectance," in *CVPR*, 2023.
[50] F. Crété-Roffet, T. Dolmiere, P. Ladret, and M. Nicolas, "The blur effect: Perception and estimation with a new no-reference perceptual blur metric," in *SPIE Electronic Imaging Symposium Conf Human Vision and Electronic Imaging*, vol. 12, 2007, pp. EI–6492.
[51] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[52] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 590–596.
[53] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.
[54] W. Ge, T. Hu, H. Zhao, S. Liu, and Y.-C. Chen, "Ref-neus: Ambiguity-reduced neural implicit surface learning for multi-view reconstruction with reflection," *arXiv preprint arXiv:2303.10840*, 2023.
[55] X. Ma, X. Xu, L. Zhang, K. Zhou, and H. Wu, "Opensvbrdf: A database of measured spatially-varying reflectance," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–14, 2023.
[56] Shining3D, "Einscan pro 2x plus handheld industrial scanner," https://www.einscan.com/handheld-3d-scanner/2x-plus/, 2023, [Online; accessed May-2023].
[57] Z. Bi, Y. Zeng, C. Zeng, F. Pei, X. Feng, K. Zhou, and H. Wu, "Gs^3: Efficient relighting with triple gaussian splatting," in *SIGGRAPH Asia 2024 Conference Papers*, 2024.

Xiang Feng is a master student in the State Key Lab of CAD & CG, Zhejiang University. He received his B.Eng. in computer science from the same university in 2022. His research interests include generation/reconstruction of appearance and shape.

Kaizhang Kang is a postdoctoral researcher at King Abdullah University of Science and Technology (KAUST). He received Ph.D. degree from Zhejiang University. His research focuses on the acquisition and reconstruction of physical information, including high-dimensional appearance, 3D geometry and volumes. He received Microsoft Asia Fellowship in 2021.

Fan Pei is a master student in the State Key Lab of CAD & CG, Zhejiang University.He received his B.Eng. from Zhejiang university in 2022.His research interests include 3D feature learning and appearance acquisition.

Huakeng Ding is an undergraduate student in the State Key Lab of CAD & CG, Zhejiang University. His research interests include appearance acquisition and shape reconstruction.

Jinjiang You is a master student in the Carnegie Mellon University. He received his B.Eng. in computer science from Zhejiang University in 2023. His research interests include appearance acquisition and 3d vision.

Ping Tan is a Professor in Department of Electronic and Computer Engineering at Hong Kong University of Science and Technology. He has previously served as the head of XR Lab at Alibaba's DAMO Academy, Chief Scientist for Computer Vision at the Artificial Intelligence Lab, and as an Associate Professor at Simon Fraser University and National University of Singapore. His research areas include computer vision and computer graphics.

Kun Zhou is a Cheung Kong Professor in the Computer Science Department of Zhejiang University, and the Director of the State Key Lab of CAD & CG. Prior to that, he was a Leader Researcher of the Internet Graphics Group at Microsoft Research Asia. He received B.S. and Ph.D. in computer science from Zhejiang University. His research interests are in visual computing, parallel computing, human computer interaction, and virtual reality.

Hongzhi Wu is a professor in the State Key Lab of CAD & CG, Zhejiang University. He received B.Sc. in computer science from Fudan University, and Ph.D. in computer science from Yale University. His current research interests include high-density illumination multiplexing devices and differentiable acquisition. Hongzhi is a recipient of Excellent Young Scholars, NSF China. He is on the editorial board of IEEE TVCG.

11 Supplementary Material

11.1 Details on Geometric Reconstruction

Once our approach convert the groups of input images into feature maps, they are directly fed as input to existing multi-view stereo techniques (be it COLMAP, NeuS or Neuralangelo), with only minor modifications: the number of channels of an input image is changed from 3 (RGB) to 12, to match the dimension of our features. The objective function stays the same as in respective reconstruction approaches. Moreover, for methods based on inverse rendering such as NeuS or Neuralangelo, we further modify their rendering process to produce 12-channel output images, in order to compute the loss against our input feature maps. This is done by changing the output dimension of the final fully connected layer in NeuS/Neuralangelo from 3 to 12.

11.2 Details on Appearance Reconstruction

After geometric reconstruction, we establish a uv-parameterization over object surfaces, and compute BRDF parameters at each valid texel via differentiable optimization. While not being tied to any specific model, we adopt the anisotropic GGX BRDF in this paper:

$$f(\omega_\mathbf{i};\omega_\mathbf{o},\mathbf{p}) = \frac{\rho_d}{\pi} + \rho_s \frac{D_{GGX}(\omega_\mathbf{h};\alpha_x,\alpha_y)F(\omega_\mathbf{i},\omega_\mathbf{h})G_{GGX}(\omega_\mathbf{i},\omega_\mathbf{o};\alpha_x,\alpha_y)}{4(\omega_\mathbf{i}\cdot\mathbf{n_p})(\omega_\mathbf{o}\cdot\mathbf{n_p})}.$$

Here ρ_d/ρ_s are the diffuse/specular albedo, α_x/α_y are the roughness parameters, and $\omega_\mathbf{h}$ is the half vector. D_{GGX} is the microfacet distribution function, F is the Fresnel term and G_{GGX} accounts for shadowing/masking effects. The BRDF model is defined in the local frame $\mathbf{n_p}/\mathbf{t_p}$ of $\mathbf{p}$, where $\mathbf{n_p}/\mathbf{t_p}$ are the normal and tangent, respectively.

To fit BRDF parameters for a particular texel, we first project its corresponding 3D position to all visible views to gather its image measurements. Next, we employ a 16D latent vector to represent the BRDF parameters: a decoder network is also trained to transform the latent vector to the parameters $(\rho_d, \rho_s, \alpha_x, \alpha_y, \mathbf{n_p}, \mathbf{t_p})$. These parameters will be used to produce rendering results, whose difference with the aforementioned image measurements is minimized. All latent vectors and the corresponding decoder are jointly optimized. Finally, we convert the latent vector at each texel to anisotropic GGX BRDF parameters, and store them in texture maps as the appearance result (as visualized in Fig. 16).

11.3 Features Incorporating Correlated Factors

According to Tab. 1, diffuse albedos and normals are mostly correlated with our learned features. Here we test the impact of replacing part of our learned features with the predictions of these highly correlated factors. Specifically, we encourage our network to learn to explicitly predict the first 6D of the output feature as diffuse albedo and normal, with the following modified loss:

$$L = \lambda_0 L_0 + \lambda_1 L_1 + \lambda_2 L_2 + \lambda_p L_p + \lambda_{\text{reg}} L_{\text{reg}},$$

where

$$L_{\text{reg}} = L_{\text{diffuse}} + L_{\text{normal}}.$$

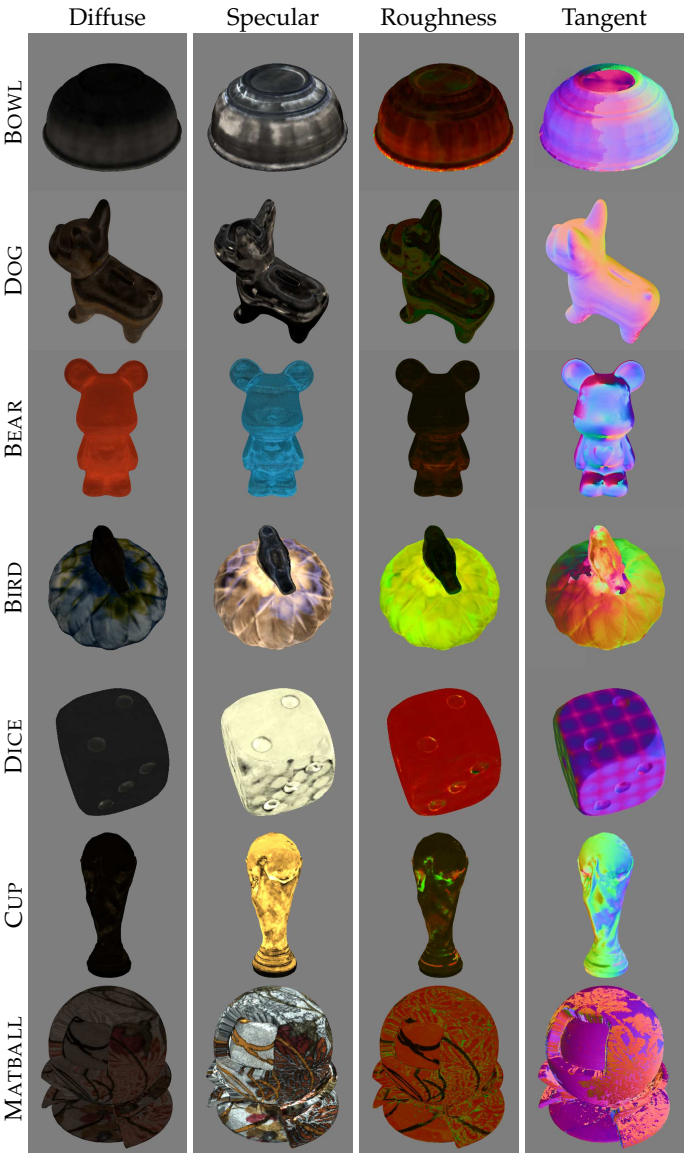

Fig. 16: Reconstructed SVBRDF parameters. For visualization purpose, each tangent is added with $(1,1,1)$ and then divided by 2 to fit to the range of $[0,1]$; the specular albedo is re-scaled; and α_x/α_y are visualized in the red/green channel.

We reserve the first 6 dimensions of the final feature for diffuse and normal predictions, and leave the remaining dimensions for data-learned features. Here L_{diffuse} represents the mean squared error (MSE) between the first three dimensions of the final feature and the ground-truth diffuse albedo, while L_{normal} is the MSE between the next three dimensions of the final feature and the ground-truth normal. We set $\lambda_{\text{reg}} = 5$ in our experiment.

We test the new features on reconstructing the geometry of Matball. Its Chamfer distance increases from 5.13 (our features) to 5.28 (new features). We find that while it is faster to train the new features due to the extra regularization term, the reconstruction quality is reduced, as the features are not completely learned from data.