Single-Image SVBRDF Estimation Using Auxiliary Renderings as Intermediate Targets

Yongwei Nie, *Member, IEEE,* Jiaqi Yu, Chengjiang Long, Qing Zhang, Guiqing Li, Hongmin Cai, *Senior Member, IEEE*

***Abstract*—Recently, single-image SVBRDF capture is formulated as a regression problem, which uses a network to infer four SVBRDF maps from a flash-lit image. However, the accuracy is still not satisfactory since previous approaches usually adopt end-to-end inference strategies. To mitigate the challenge, we propose "*auxiliary renderings*" as the intermediate regression targets, through which we divide the original end-to-end regression task into several easier sub-tasks, thus achieving better inference accuracy. Our contributions are threefold. First, we design three (or two pairs of) auxiliary renderings and summarize the motivations behind the designs. By our design, the auxiliary images are bumpiness-flattened or highlight-removed, containing disentangled visual cues about the final SVBRDF maps and can be easily transformed to the final maps. Second, to help estimate the auxiliary targets from the input image, we propose two mask images including a bumpiness mask and a highlight mask. Our method thus first infers mask images, then with the help of the mask images infers auxiliary renderings, and finally transforms the auxiliary images to SVBRDF maps. Third, we propose backbone UNets to infer mask images, and gated deformable UNets for estimating auxiliary targets. Thanks to the well-designed networks and intermediate images, our method outputs better SVBRDF maps than previous approaches, validated by the extensive comparisonal and ablation experiments.**

***Index Terms*—SVBRDF, auxiliary rendering, bumpiness, highlight.**

I. Introduction

OBJECTS are comprised of diverse materials, such as metal, plastic, stone, and wood, etc., exhibiting varying appearances due to their distinctive reflectance properties. For opaque materials, reflectance is typically modeled by a 6D Spatially-Varying Bidirectional Reflection Distribution Function (SVBRDF), e.g., Cook-Torrance model [1] which characterizes materials by four SVBRDF maps: surface normal $\mathbf{N}$, diffuse albedo $\mathbf{D}$, roughness $\mathbf{R}$, and specular albedo $\mathbf{S}$. The Cook-Torrance SVBRDF model establishes a rendering equation that can generate a visually-realistic image $\mathbf{I}$ given the four SVBRDF maps in conjunction with a pair of lighting and viewing directions:

$$\mathbf{I} = \mathcal{R}\left((\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S}), (\mathbf{l}, \mathbf{v})\right). \tag{1}$$

where $\mathbf{l}$ and $\mathbf{v}$ are lighting and viewing directions respectively, and $\mathcal{R}$ is the rendering function. However, despite the straightforward nature of the forward rendering in Eq. 1, the inverse problem of estimating the four SVBRDF maps from an image has long been a challenging issue.

Previous research [2; 3; 4; 5] attempt to sample the reflectance function of material with physical equipments, suffering from the time complexity of the sampling. Approaches of [6; 7; 8; 9; 10; 11; 12] simplify the sampling procedure by assuming strong priors, which may fail whenever the assumptions are not satisfied. Recently, Ma et al. [13] proposed a high-quality reflection acquisition system consisting of 2 cameras and 16,384 LEDs, which greatly reduces the time of the physically capturing to 15 minutes per sample.

On the other hand, deep networks have been employed to infer SVBRDF parameters from as few as a single image. Many of these methods are regression-based, designing and training neural networks to output the four SVBRDF maps given an input image. For example, Deschaintre et al. [14] augmented the UNet [15] with a global branch for single-image SVBRDF capture. Li et al. [16] utilized an encoder-decoder style network improved by incorporating a classifier in the middle to process different material types separately. Guo et al. [17] proposed a highlight-aware convolution operation to effectively handle highlights within an image.

We observe that the aforementioned methods are end-to-end approaches, i.e., they directly infer the target material maps given the input image. Since the appearance of the target images are very different from that of the input image (e.g., the normal map looks very different from the corresponding input image), we argue that it is beneficial to introduce guidance into the end-to-end inference pipeline, providing regularization to the challenging task. Let $\mathbf{I} \rightarrow (\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$ represents the end-to-end regression task that outputs $(\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$ from the input $\mathbf{I}$. We adopt the intermediate-target-guided paradigm of:

$$\mathbf{I} \rightarrow \mathbf{A} \rightarrow (\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S}), \tag{2}$$

where $\mathbf{A}$ represents the intermediate targets. This idea was originally exploited in the work of MaterIA [18] which first estimates an irradiance map and then uses it to assist the estimation of the normal map. We go further in this paper to apply this strategy to all the four material components. The main contribution of this paper is the proposition of a set of meaningful and useful intermediate targets.

To make our method effective, the two sub-tasks of $\mathbf{I} \rightarrow \mathbf{A}$ and $\mathbf{A} \rightarrow (\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$ should be easier than the original end-to-end task. That means, $\mathbf{A}$ should be similar to the input $\mathbf{I}$, while at the same time being able to be transformed to $(\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$ easily. To meet the two requirements, we propose to rely on the forward rendering function $\mathcal{R}$ in Eq. 1 to

Yongwei Nie, Jiaqi Yu, Guiqing Li, Hongmin Cai are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong, China.

Chengjiang Long is with the Meta Reality Labs, USA.

Qing Zhang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong, China.

render images of $\mathbf{A}$, which we term as "auxiliary renderings" (or "aux-renderings"). As the aux-renderings are generated by the same rendering function as the input image and share some material components with it, they tend to look similar to the input image to a certain extent. We propose a total of three aux-renderings. In conjunction with the input image, they form two pairs of images that contain visual cues about the target SVBRDF maps, such that they can be easily transformed to the SVBRDF maps. The proposed aux-renderings are called bumpiness-flattened and highlight-removed auxiliary images. For more detailed information about them, as well as the rationale and considerations behind our designs, please refer to the main text.

Although the proposed aux-renderings show similar appearance to the input image, the difference between them is still prominent. For instance, the highlight-removed aux-renderings do not have highlights while the input image does. To further facilitate the inference of aux-renderings from the input image, we propose two mask images, namely a bumpiness mask and a highlight mask, achieved by modifying the rendering function $\mathcal{R}$. Ultimately, our method comprises three stages:

$$\mathbf{I} \rightarrow \mathbf{M} \rightarrow \mathbf{A} \rightarrow (\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S}), \tag{3}$$

i.e., inferring mask images from an input image, then inferring aux-renderings with the aid of the mask images, and finally inferring SVBRDF maps from the input and auxiliary images, where $\mathbf{M}$ represents the mask images.

We design different networks to fulfill tasks at different stages. For mask images, we design backbone UNets that leverage EfficientNet-B3 [19] as the backbone network, which owns the ability to recognize low/highlight regions in images due to its pre-training on extensive datasets. For aux-renderings, we propose gated deformable UNets with higher inpainting capabilities to remove bumpiness and highlight effects in images. With the effort on the intermediate images and networks design, we can finally rely on a simple UNet from [15] to estimate SVBRDF maps from an input image.

In summary, the contributions of our paper are:

- We propose a multi-stage single-image SVBRDF estimation method with aux-renderings as intermediate targets, and we propose a total of three aux-renderings.
- To help infer the aux-renderings from an input image, we further propose two mask images, including a bumpiness mask and a highlight mask, obtained by modifying the forward rendering function.
- We propose backbone-UNets for inferring the mask images, and gated deformable UNets for inferring the aux-renderings, which output satisfactory inverse inference results as validated by the experiments.

II. Related Work

A. Material Models and Traditional Acquisition Methods

There are different kinds of materials in this world, for which different representation models have been proposed, e.g., BSSRDF [20], BTDF [21], and BRDF [22], etc. Among them, BRDF assumes light hitting a surface point leaves the surface at the same point, which is suitable for modeling materials with nearly flat and opaque surfaces [23]. In this paper, we are interested in spatially-varying BRDF (SVBRDF). Compared with BRDF that represents reflectance characteristics of homogeneous materials, SVBRDF is an extension of BRDF to non-homogeneous materials. One of the most famous SVBRDF models is the Cook-Torrance model [1], the reflection function of which is defined on four material properties including surface normal, diffuse albedo, roughness, and specular albedo. Our work is based on the Cook-Torrance model and tries to estimate the four spatially-varying reflectance parameters given only one image.

Traditional SVBRDF capture approaches usually require hardware setups to spatially or angularly sample a material from many different lighting and viewing directions [7; 12; 24; 25; 26; 27; 28; 29; 30; 31]. The cumbersome equipments prevent novice users from using these sophisticated approaches, hindering the widespread application of these methods. Many approaches use affordable mobile phones or RGB-D cameras for SVBRDF capture [8; 11; 32; 33; 34; 35; 36], but usually assume priors on the target materials. For example, the work of [8] assumes repetitive textures as input. The methods of [10; 11; 12; 37; 38] simplify the measured BRDFs to be approximately represented by a linear combination of basis functions such as spherical harmonics or wavelets. These assumptions limit the kinds of materials they can handle. This line of research is constantly developing. For example, Ma et al. [13] proposed a new capturing system consisting of 2 cameras and 16,384 LEDs which can capture high-quality SVBRDF in less than 15 minutes. Using their system, they collect a dataset of 1000 SVBRDFs from real images.

B. Deep Learning-based SVBRDF Capture

Probably Aittala et al. [39] were the first adopting neural networks for texture SVBRDF capture, measuring the distance between texture patches based on networks. Li et al. [40] (further improved by [41]) proposed the first neural network that directly regresses SVBRDF maps from an image with a dataset self-augmenting strategy. Works of [14] and [16] contribute to the community large-scale synthetic datasets. Based on the datasets, Deschaintre et al. [14] augmented the UNet [15] with a global branch for single-image SVBRDF capture. Li et al. [16] adopted an encoder-decoder network incorporating with a classifier. Guo et al. [17] proposed a highlight-aware convolution operation to handle highlights in input images. Henzler et al. [42] proposed a method first converting photos into latent material codes and then generating infinite BRDF model parameters conditioned on the codes. Wang et al. [43] leveraged the assumption that a material is a linear combination of a set of basis materials. They proposed a nice two-level model for estimating the basis materials and their corresponding weights. The mentioned approaches perform end-to-end inverse SVBRDF estimation. Differently, we design auxiliary renderings as intermediate targets in the middle of the SVBRDF recovery process.

The above approaches admit a single image as input, while there are methods requiring multiple images as input. In

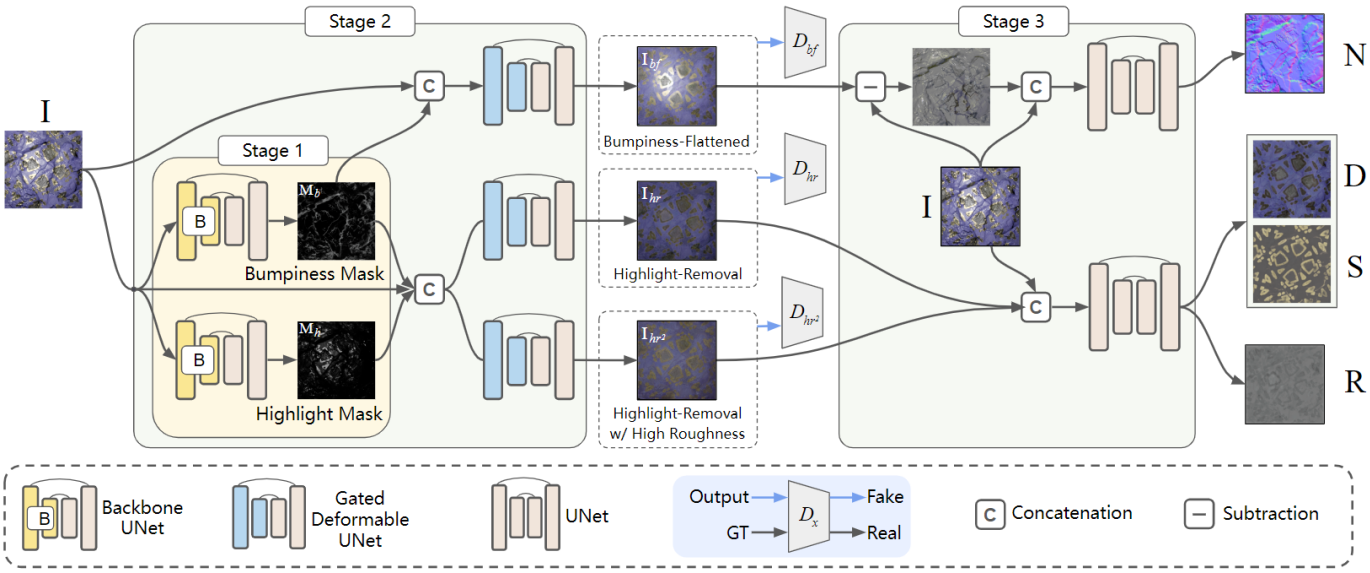

Fig. 1. Given an image $\mathbf{I}$, our method estimates SVBRDF maps (i.e., $\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S}$) in three stages. First, we generate two mask images $\mathbf{M}_b$ and $\mathbf{M}_h$ (denoting *b*umpiness mask and *h*ighlight mask) from the input image $\mathbf{I}$, using the proposed backbone UNets. Then with the help of the mask images, we generate three auxiliary images $\mathbf{I}_{bf}$, $\mathbf{I}_{hr}$, and $\mathbf{I}_{hr^2}$ in the second stage by gated deformable UNets. Finally in the third stage, SVBRDF maps are inferred from combinations of the input and auxiliary images, fulfilled by UNets in its original form [15]. Please see the main text for more detailed pipeline of the method.

[44], the single-image method of [14] is extended to handle multiple images. In [45], the obtained SVBRDF parameters are further transferred to large-scale objects of similar appearance. Instead of regression, methods of [46; 47; 48] exploit optimization-based approaches. They train latent SVBRDF parameter spaces and optimize in the space to find a solution that best reproduces observations. Zhou et al. [49] and Fischer et al. [50] employ meta-learning to combine the advantages of both regression and optimization approaches. Fan et al. [51] proposed a reflection acquisition method from two images captured by a wide-angle lens and a zoom lens of a smartphone.

Besides the above image-to-image translation models, there are also methods for conditional/unconditional general of materials [52; 53; 54]. For example, Vecchio et al. [55] used discriminators to minimize the distribution difference between SVBRDF maps of real and synthetic images. Zhou et al. [56] recovered SVBRDF maps of one real image, and the light position of another real image. Then the re-rendered image using SVBRDF maps of the first image and the light of the second image was compared with the second real image by a discriminator. Works of [57] and [58] recover SVBRDFs maps in purely unsupervised ways, where Zhao et al. [57] recovered and synthesized large SVBRDF maps jointly, while the method of [58] was designed to process stationary materials using a Fourier coefficient-based loss function.

There are emerging works [18; 58; 59; 60; 61; 62; 63] on tileable and controllable material generation, on high-resolution SVBRDF capture, and on material transfer. Among them, MaterIA [18] observes that some material components are easier to estimate than others. Therefore, the easier components can be estimated at first and then be used to assist the estimation of the other components. For example, MaterIA first computes irradiance, and then computes the normal map from the combination of the input and irradiance images. Different from our approach, MaterIA only applies this strategy to the estimation of the normal map, while we design two mask images and three auxiliary maps to assist the estimation of all kinds of material components. Besides, the recent work [64] optimizes planar lighting pattern, while our method works under point light source. Compared to our method that handle nearly planar surfaces, works of [35; 36; 65; 66; 67] recover both shape and SVBRDFs of more complex scenes (such as shiny 3D cultural relics [67]).

III. Our Method

A. Overview

Figure 1 shows the pipeline of our method. We recover four SVBRDF maps $(\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$ from a single flash image $\mathbf{I}$ that captures a nearly planar surface lit by a flash, according to the Cook-Torrance model [1]. Our method is composed of three stages. First, we propose "backbone UNets" to extract bumpiness and highlight masks (denoted as $\mathbf{M}_b$ and $\mathbf{M}_h$) from $\mathbf{I}$. Second, we design gated deformable UNets to generate three aux-renderings including bumpiness-flattened image $\mathbf{I}_{bf}$, and highlight-removed images $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$. Third, the aux-renderings together with the input image are transformed to SVBRDF maps by UNets [15].

In the following, we first show how we prepare the intermediate images, and the motivations behind the designs. Then, we describe the details of the backbone UNet and gated deformable UNet that compose our network. Finally, we introduce losses used to train our model.

B. Auxiliary Renderings

Our method assumes a dataset that is composed of tuples of $(\mathbf{I}, \mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$, where $\mathbf{I}$ is an input image, and $\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S}$ are the ground-truth SVBRDF maps of the input image. We use

Fig. 2. Examples of $\mathbf{I}$ and $\mathbf{I}_{bf}$ where $\mathbf{I}$ is rendered with normal map $\mathbf{N}$ while $\mathbf{I}_{bf}$ is rendered with $\mathbf{N}_u$. Visually, $\mathbf{I}$ exhibits bumpiness effect matching the spatial variation in $\mathbf{N}$. Since $\mathbf{N}_u$ is smooth, the corresponding image $\mathbf{I}_{bf}$ is free of bumpiness effect.

the following ways to further generate three auxiliary images used for training our method.

Bumpiness-Flattened Auxiliary Rendering. $\mathbf{I}_{bf}$ is produced by:

$$\mathbf{I}_{bf} = \mathcal{R}\left((\mathbf{N}_u, \mathbf{D}, \mathbf{R}, \mathbf{S}), (\mathbf{l}, \mathbf{v})\right). \tag{4}$$

Compared with Eq. 1, the only difference is that the original normal map $\mathbf{N}$ is replaced with a uniform normal map $\mathbf{N}_u$. In our setup, every pixel of the uniform normal map $\mathbf{N}_u$ is filled with $(0.5, 0.5, 1)$. Figure 2 gives examples of $\mathbf{I}$ and $\mathbf{I}_{bf}$. As can be seen, $\mathbf{I}$ contains bumpiness effect caused by the normal variation in $\mathbf{N}$, while $\mathbf{I}_{bf}$ does not since $\mathbf{N}_u$ used to generate it is smooth everywhere.

Highlight-Removed Auxiliary Rendering. $\mathbf{I}_{hr}$ is obtained by computing the mean of a set of $\mathbf{I}_{bf}$:

$$\mathbf{I}_{hr} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{I}_{bf}^{k}. \tag{5}$$

where $K = 100$ is the number of $\mathbf{I}_{bf}$, and the k^{th} one is obtained by:

$$\mathbf{I}_{bf}^{k} = \mathcal{R}\left((\mathbf{N}_u, \mathbf{D}, \mathbf{R}, \mathbf{S}), (\boldsymbol{l}_k, \boldsymbol{v})\right). \tag{6}$$

All $\mathbf{I}_{bf}$ share the same $\mathbf{N}_u$, $\mathbf{D}$, $\mathbf{R}$, $\mathbf{S}$, and $\boldsymbol{v}$, but are rendered under different lighting conditions $\boldsymbol{l}_k$ (please refer to the supplemental material for the lighting scheme).

Figure 3 shows examples of $\mathbf{I}_{bf}$ and the corresponding $\mathbf{I}_{hr}$. It can be seen that in different $\mathbf{I}_{bf}$, different regions are illuminated. Thanks to the average operator in Eq. 5, the highlight effect is removed from $\mathbf{I}_{hr}$. That is why we call $\mathbf{I}_{hr}$ a highlight-removed image.

Highlight-Removed Auxiliary Rendering with High Roughness. The generation process of $\mathbf{I}_{hr^2}$ is very similar to that of $\mathbf{I}_{hr}$. First, we generate different temporary images:

$$\mathbf{I}_{tmp}^{k} = \mathcal{R}\left((\mathbf{N}_u, \mathbf{D}, \mathbf{R}_1, \mathbf{S}), (\boldsymbol{l}_k, \boldsymbol{v})\right). \tag{7}$$

Compared with Eq. 6, besides replacing $\mathbf{N}$ with $\mathbf{N}_u$, we further replace the roughness map $\mathbf{R}$ with $\mathbf{R}_1$ which is filled

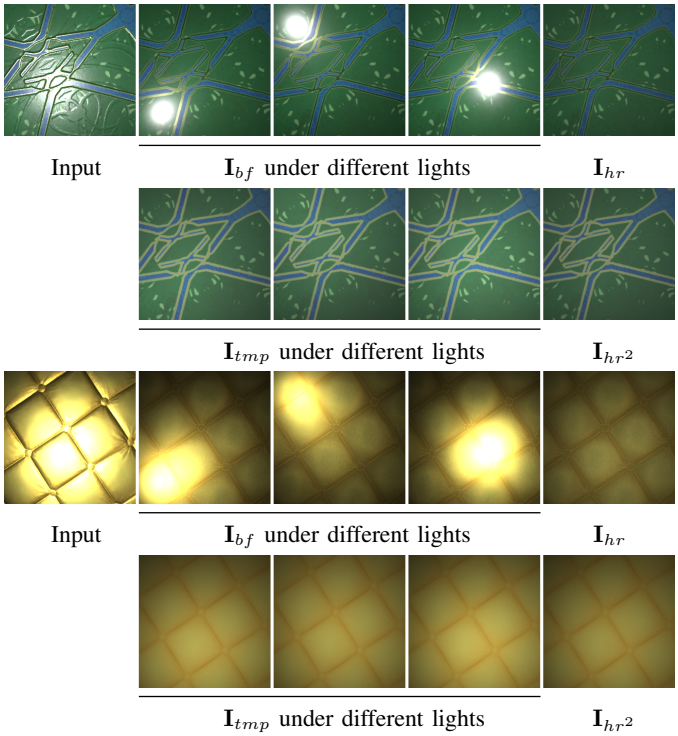

Fig. 3. Top row of each example shows $\mathbf{I}_{bf}$ lit by different lightings and the corresponding $\mathbf{I}_{hr}$ by averaging all $\mathbf{I}_{bf}$, and bottom row shows $\mathbf{I}_{tmp}$ lit by different lightings and the corresponding $\mathbf{I}_{hr^2}$ by averaging all $\mathbf{I}_{tmp}$.

with the highest roughness 1.0 everywhere. Then, $\mathbf{I}_{hr^2}$ is obtained by averaging all the temporary images:

$$\mathbf{I}_{hr^2} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{I}_{tmp}^{k}. \tag{8}$$

Similarly, we show examples of $\mathbf{I}_{tmp}$ and the corresponding $\mathbf{I}_{hr^2}$ in the bottom row of Figure 3. Since the roughness is very high in $\mathbf{I}_{tmp}$, the highlight effect in $\mathbf{I}_{tmp}$ is weaker than that in the corresponding $\mathbf{I}_{bf}$. By averaging all $\mathbf{I}_{tmp}$, the highlight effect is further removed from the obtained $\mathbf{I}_{hr^2}$.

C. Motivations Behind the Designs

Now, we introduce the philosophy behind the design of the above three auxiliary images.

Generating Auxiliary Renderings Computationally Instead of Physically. We generate all the auxiliary renderings in computational manners rather than finding physically meaningful renderings as intermediate targets. That is because it is convenient to modify the input parameters of the forward SVBRDF rendering function to obtain different rendered images, while finding a physically meaningful intermediate target requires professional material design and use experience. In addition, since there are many alternatives for the input parameters and their combinations, the design space of the computational way is large, and there are more chances to find qualified intermediate targets. In comparison, physically meaningful renderings are rare and provide far fewer options.

Relying on the Forward Rendering Function for the Generation. We always rely on the forward rendering function

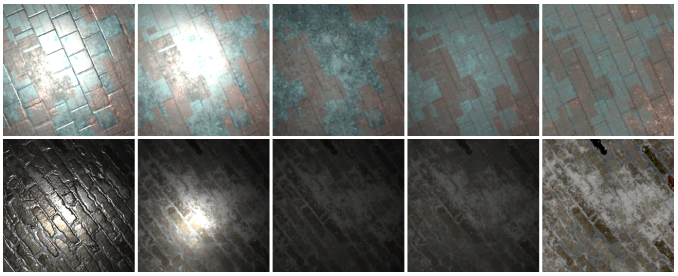

Fig. 4. From left to right: input image $\mathbf{I}$, the corresponding bumpiness-flattened image $\mathbf{I}_{bf}$, highlight-removed images $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$, and the visualization of the sum of diffuse ($\mathbf{D}$) and specular ($\mathbf{S}$) maps.

to generate aux-renderings, e.g., $\mathbf{I}_{bf}$, $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$ are obtained based on the rendering functions in Eq. 4, Eq. 6 and Eq. 7, respectively. Since both the input and aux-renderings are generated through the forward rendering equation, the differences between them are only caused by the changed SVBRDF parameters. The remaining material parameters and the shared lighting and viewing directions allow the aux-renderings to maintain similar appearance to the input image.

Figure 4 shows examples of input images $\mathbf{I}$ and the corresponding $\mathbf{I}_{bf}$, $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$. Taking $\mathbf{I}$ and $\mathbf{I}_{bf}$ as examples, although the normals are flattened, $\mathbf{I}_{bf}$ exhibits similar object structure and textures to $\mathbf{I}$ due to the remaining unchanged material parameters. Since more parameters are changed when rendering $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$, they are farther away from the input image in appearance. However, we can still observe similar elements between $\mathbf{I}$ and $\mathbf{I}_{hr}$ (or $\mathbf{I}_{hr^2}$). These connections between the input and aux-renderings reduce the difficulty of inferring the aux-renderings from the input image.

One may find that $\mathbf{I}_{hr^2}$ may be theoretically similar to the sum of diffuse and specular maps (i.e., $\mathbf{D}+\mathbf{S}$), since they are both computed based on the diffuse and specular maps (see the top row of Figure 4). However, since $\mathbf{I}_{hr^2}$ is generated through the forward rendering function, it is further affected by the rendering process and the lighting and viewing directions. Compared with $\mathbf{D}+\mathbf{S}$, $\mathbf{I}_{hr^2}$ has more similar light/shadow effects as the input image (see the bottom row of Figure 4. Ablation studies in Section IV-E5 also demonstrate that it is easier to generate $\mathbf{I}_{hr^2}$ than $\mathbf{D}+\mathbf{S}$ given an input image.

Using Pairs of Renderings as Intermediate Targets Instead of Only One. At first glance, one may be confused by the too many auxiliary renderings we propose. In fact, we just propose two pairs of renderings. They are $(\mathbf{I}, \mathbf{I}_{bf})$ and $(\mathbf{I}_{hr}, \mathbf{I}_{hr^2})$, where $\mathbf{I}$ and $\mathbf{I}_{bf}$ are different in the parameter of normal, while $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$ differ in roughness. Our core idea is to use a pair of renderings instead of only one as the intermediate targets to infer SVBRDF maps, leveraging the difference between the pair of renderings. We use $(\mathbf{I}, \mathbf{I}_{bf})$ to infer the normal map. $\mathbf{I}$ contains bumpiness effect caused by the normal map while $\mathbf{I}_{bf}$ does not (see Figure 2). Feeding them together to a neural network can help the network identify the difference between them to output the normal map.

We use $(\mathbf{I}_{hr}, \mathbf{I}_{hr^2})$ to infer the remaining diffuse, roughness and specular maps. First, they are both rendered with the uniform normal map. Using them as intermediate targets, we avoid the influence of normals when inferring the three maps. Second, since $\mathbf{I}_{hr}$ is rendered using the original roughness map, it contains appearance changes caused by spatially varying roughness (see the correlation between $\mathbf{I}_{hr}$ and $\mathbf{R}$ in the top row of Figure 5), while $\mathbf{I}_{hr^2}$ does not because it is rendered with the same roughness everywhere (see $\mathbf{I}_{hr^2}$ in the same row). This allows the network to infer the roughness map from the difference between $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$. Third, we find that the roughness essentially influences the effect of specular albedo. High specular material usually has low roughness and looks dark when *not* viewing from the mirror reflection direction (see the zoomed-in part of $\mathbf{I}_{hr}$ of the second example of Figure 5). By setting the high-specular material with the roughness of 1.0, it is rendered to be brighter than before (see the same part of $\mathbf{I}_{hr^2}$). This difference facilitates the network to infer the specular map from $(\mathbf{I}_{hr}, \mathbf{I}_{hr^2})$. Finally, we use $(\mathbf{I}_{hr}, \mathbf{I}_{hr^2})$ to infer the diffuse map too, for simplicity.

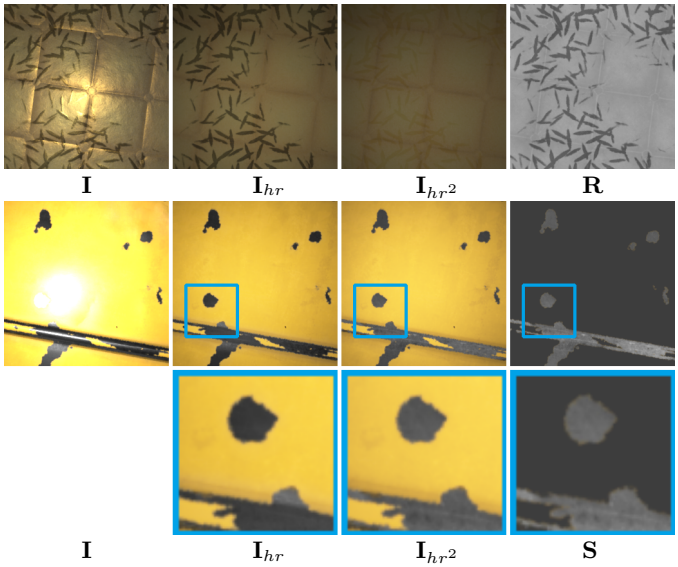

Fig. 5. Top row shows an example where $\mathbf{I}_{hr}$ is highly correlated with the roughness map $\mathbf{R}$, while $\mathbf{I}_{hr^2}$ is not since it is rendered with the highest roughness everywhere. Bottom row shows an example of how roughness affects the rendering effect of specular albedo: high specular (see $\mathbf{S}$) looks dark (see $\mathbf{I}_{hr}$), but becomes brighter (see $\mathbf{I}_{hr^2}$) after setting a higher roughness.

We have also attempted to use the way processing $\mathbf{N}$ to estimate the other three maps, which however fails. Please refer to the supplemental material for more interpretations.

Handling Highlights. If there is no highlight, $\mathbf{I}_{bf}$ and $\mathbf{I}_{tmp}$ suffice to infer the diffuse, specular and roughness maps. However in our setting, the input image is lit by flash. Highlights help reveal specular and roughness components but make a part of pixels completely saturated. Previous approaches attempt to repair the saturated region with improved convolutional operators [17]. Differently, we employ the highlight-free images $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$ as intermediate targets to explicitly guide the model to inpaint the highlight regions.

Toy Experiment. To validate the effectiveness of the proposed auxiliary renderings, we conduct a toy experiment that directly transforms the ground-truth aux-renderings to SVBRDF maps using UNets [15]. Please see Section IV-E1 for the upper-bound accuracy our method can achieve.

D. Bumpiness and Highlight Masks

We have proposed the aux-renderings and demonstrated they can be conveniently transformed to SVBRDF maps. The remaining question is how to estimate high-quality aux-images from the input image. In fact, estimating $\mathbf{I}_{bf}$ from $\mathbf{I}$ is easy as $\mathbf{I}_{bf}$ looks similar to $\mathbf{I}$. The only difference between them is the bumpiness to be removed. However, it is much challenging to obtain $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$, as besides the bumpiness effect we also need to remove the highlight effect. Inspired by approaches in the field of shadow/highlight removal [68; 69] which prove that taking shadow/highlight mask can improve the shadow/highlight removal effect, we propose bumpiness mask $\mathbf{M}_b$ and highlight mask $\mathbf{M}_h$ to help remove the bumpiness and highlight effect from $\mathbf{I}$.

Bumpiness mask can be approximated by lowlight mask, as low light usually appears in concave regions. We generate both lowlight (or bumpiness) and highlight masks by modifying the Cook-Torrance forward rendering function. Recall that the definition of the Cook-Torrance BRDF reflection model is:

$$\rho(\mathbf{n}_j, \mathbf{d}_j, r_j, \mathbf{s}_j, \mathbf{l}_j, \mathbf{v}_j) = \frac{\mathbf{d}_j(1-\mathbf{s}_j)}{\pi} + \frac{D(\mathbf{n}_j, r_j)F(\mathbf{s}_j)G(\mathbf{n}_j, r_j)}{4(\mathbf{n}_j \cdot \mathbf{v}_j)(\mathbf{n}_j \cdot \mathbf{l}_j)}, \quad (9)$$

where $\mathbf{n}_j$, $\mathbf{d}_j$, r_j, and $\mathbf{s}_j$ are the normal, diffuse, roughness and specular values at the j^{th} pixel of the four SVBRDF maps $\mathbf{N}$, $\mathbf{D}$, $\mathbf{R}$, and $\mathbf{S}$, respectively; $\mathbf{v}_j$ and $\mathbf{l}_j$ are viewing and lighting directions. The first term in the above equation defines diffuse reflectance while the second term models specular reflectance. We observe that among the three functions $D(\cdot)$, $F(\cdot)$ and $G(\cdot)$ in the numerator of the second term, the normal distribution function $D(\cdot)$ strongly controls the intensity of the reflected light. If $D(\cdot)$ returns a large value, that means strong specularity occurs and highlight is observed. Otherwise, a small amount of light is reflected, creating dark pixels. Although $F(\cdot)$ and $G(\cdot)$ also affect the strength of the reflection, their effect takes place more at grazing angles, while in our situation the view angle is always 90° perpendicular to the captured surface which suppresses the effects of both functions.

To generate highlight mask $\mathbf{M}_h$, we thereby suppress the high response of $D(\mathbf{n}_j, r_j)$ by:

$$D_h(\mathbf{n}_j, r_j) = \begin{cases} \bar{D} & , \text{if } D(\mathbf{n}_j, r_j) > \delta_h, \\ D(\mathbf{n}_j, r_j), & otherwise, \end{cases} \quad (10)$$

where δ_h is a threshold which is 0.9 in this paper (please refer to the supplemental material for more discussions about the threshold selection process), and $\bar{D}$ is the average of $D(\mathbf{n}_j, r_j)$ of pixels whose $D(\mathbf{n}_j, r_j)$ is smaller than δ_h. This modified normal distribution function D_h yields an image with less strong highlight. We then subtract it from the input image to get the highlight mask $\mathbf{M}_h$. Similarly, to obtain bumpiness mask $\mathbf{M}_b$, we compensate the low-response of $D(\mathbf{n}_j, r_j)$ by:

$$D_l(\mathbf{n}_j, r_j) = \begin{cases} \delta_s & , \text{if } D(\mathbf{n}_j, r_j) < \delta_s, \\ D(\mathbf{n}_j, r_j), & otherwise, \end{cases} \quad (11)$$

where δ_s is 0.15. This normal distribution function D_l renders an image without too lowlight pixels, from which we subtract the input image to obtain the bumpiness mask $\mathbf{M}_b$.

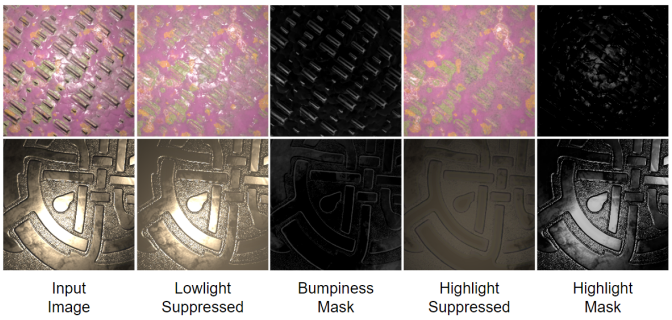

Fig. 6. From left to right: input image, lowlight-suppressed image, bumpiness mask $\mathbf{M}_b$, highlight-suppressed image, and highlight mask $\mathbf{M}_h$.

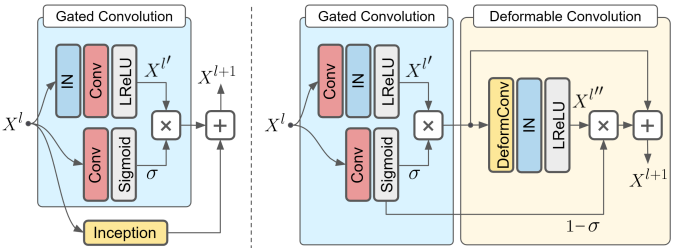

Fig. 7. Left: Highlight-aware convolution (HA-Conv) from [17]. It contains a gated convolution module and an inception module. The two modules are placed in parallel. Right: GAted deformable convolution (GAD-Conv) proposed in this paper consisting of a gated convolution module and a deformable convolution module which are sequentially executed.

Figure 6 shows examples of the bumpiness and highlight mask images. From left to right are input images, lowlight-suppressed image, bumpiness mask, highlight-suppressed image, and highlight mask. As can be seen, the two kinds of mask images accurately indicate the bumpiness and highlight regions of the input images.

E. Networks

Now, we introduce the networks implementing our method. Detailed architectures are in the supplemental material.

Phase 1: Estimating Mask Images using "backbone" UNets. In the first stage, we estimate bumpiness and highlight masks from the input image. To fulfill this task, we employ the paradigm of transfer learning. That is, we use EfficientNet-B3 [19] as the backbone to extract features from the input image. The extracted features are then transformed to the mask images by a decoder network. During training, the backbone network is only finetuned with a small learning rate.

EfficientNet-B3 has 9 stages that progressively extract features with gradually decreasing spatial resolution but increasing channel size. We consider the features at stage 2, 3, 4, 6, and 7, and use a sequence of standard convolution blocks to parse these features and gradually upsample them until the spatial and channel resolution of the mask image is reached. EfficientNet-B3 can be viewed as an encoder. The standard convolution blocks compose a decoder. We add skip connections between them to obtain a UNet-shaped network.

Phase 2: Estimating Auxiliary Renderings by GAted Deformable UNet (GAD-UNet). Next, with the mask images, we estimate auxiliary renderings from the input image. We

simply concatenate the input $\mathbf{I}$ and bumpiness mask $\mathbf{M}_b$ to generate the bumpiness-flattened image $\mathbf{I}_{bf}$, while for the two highlight-removed images $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$, we concatenate $\mathbf{I}$, $\mathbf{M}_b$, and $\mathbf{M}_h$ as input. We design a GAted Deformable Convolutional module (GAD-Conv) to fulfill the above tasks, inspired by the Highlight-Aware Convolution (HA-Conv) proposed in [17]. Figure 7 (a) illustrates HA-Conv which consists of a gated convolution branch and an inception branch. Let $\mathbf{X}^l$ be the input, the gated convolution module uses one branch to compute features $\mathbf{X}^{l'}$, and the other branch to output the gating variable σ. The output of the gated convolution module is $\sigma \cdot \mathbf{X}^{l'}$, where the dot means element-wise multiplication. By the gating variable, the gated convolution module learns to ignore undesirable features of highlight (or bumpiness) regions. However, the inception block of HA-Conv may re-add the undesirable features back into the network (see the ablation in Section IV-E4).

We improve HA-Conv by removing the inception branch but appending a deformable convolution block after the gated convolution block, proposing GAD-Conv as shown in Figure 7 (b). In GAD-Conv, the output of the gated convolution, i.e., $(\sigma \cdot \mathbf{X}^{l'})$, is further processed by a deformable convolution block [70]. Denote the output of the deformable convolution block by $\mathbf{X}^{l''}$. The final output of the GAD-Conv is:

$$\mathbf{X}^{l+1} = \sigma \cdot \mathbf{X}^{l'} + (1-\sigma) \cdot \mathbf{X}^{l''}. \tag{12}$$

In short, by GAD-Conv, we first remove unwanted features of bumpiness and highlight regions by the gated convolution, then inpaint these regions by the deformable convolution. Since the deformable convolution has larger receptive field, it can leverage content far away from the bumpiness/highlight regions to inpaint the saturated information.

Finally, we propose the GAD-UNet used to generate the auxiliary renderings. GAD-UNet is similar to the original UNet model of [15], except that we adopt GAD-Conv instead of the standard convolution module in the encoder.

Phase 3: Estimating SVBRDF maps by UNets. Finally, we use the original UNet of [15] to estimate SVBRDF maps from the input image and the aux-renderings, except for differences in hyperparameters (see the supplemental material). For the normal map, instead of directly concatenating $\mathbf{I}$ and $\mathbf{I}_{bf}$ as input, we first compute the difference between $\mathbf{I}$ and $\mathbf{I}_{bf}$, and then use the concatenation of $\mathbf{I}$ and $(\mathbf{I}-\mathbf{I}_{bf})$ as input. For the other three maps, we directly concatenate $\mathbf{I}$, $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$ as input. We infer the diffuse map $\mathbf{D}$ and specular map $\mathbf{S}$ simultaneously, as they are more closely related to each other. We use a separate UNet to estimate the roughness map $\mathbf{R}$.

Discriminator. For each of the auxiliary renderings, we apply the WGAN loss [71]. The adopted discriminator network is also borrowed from [71].

F. Training Losses

Loss for Mask Images. For mask images, we only apply the following L_2 reconstruction loss:

$$\mathcal{L}_{Mask} = \|\mathbf{M} - \hat{\mathbf{M}}\|_2, \tag{13}$$

where $\mathbf{M}$ and $\hat{\mathbf{M}}$ stand for the ground-truth and estimated mask images, respectively.

Losses for Auxiliary Renderings. Let $\mathbf{X}$ be one of $\mathbf{I}_{bf}$, $\mathbf{I}_{hr}$, $\mathbf{I}_{hr^2}$. Each auxiliary rendering is supervised by a L_1 loss, a WGAN loss, and a perceptual loss. The L_1 loss is:

$$\mathcal{L}_{Rec} = \|\mathbf{X} - \hat{\mathbf{X}}\|_1, \tag{14}$$

where $\hat{\mathbf{X}}$ denotes the estimated auxiliary rendering. The WGAN loss is:

$$\begin{aligned}\mathcal{L}_{Adv} =& \min_G \max_D \mathop{\mathbb{E}}_{\mathbf{X}\sim\mathbb{P}_s}[D(\mathbf{X})] - \mathop{\mathbb{E}}_{\hat{\mathbf{X}}\sim\mathbb{P}_g}[D(\hat{\mathbf{X}})] \\ &+\lambda \mathop{\mathbb{E}}_{\tilde{\mathbf{X}}\sim\mathbb{P}_{\tilde{\mathbf{X}}}}[(\left\|\nabla_{\tilde{\mathbf{X}}} D(\tilde{\mathbf{X}})\right\|_2 - 1)^2],\end{aligned} \tag{15}$$

where G represents the network producing $\hat{\mathbf{X}}$, and D represents the discriminator network, $\mathbb{P}_s$ is the distribution of $\mathbf{X}$, and $\mathbb{P}_g$ stands for the distribution of the estimated images, $\tilde{\mathbf{X}}$ is an interpolation between $\mathbf{X}$ and $\hat{\mathbf{X}}$. Finally, λ is a weight to balance the two terms. We use $\lambda = 10$ during training. Finally, the perceptual loss is defined as:

$$\mathcal{L}_{Percep} = \frac{1}{N}\sum_{i=0}^{N} \|\Phi^i(\mathbf{X}) - \Phi^i(\hat{\mathbf{X}})\|_1, \tag{16}$$

where $\Phi^i(\cdot)$ stands for the feature maps of the i-th pooling layer of the pre-trained VGG16 network. We use $N = 3$ layers here, which are *pool*-1, *pool*-2, and *pool*-3 of VGG16. The total loss for an auxiliary rendering is:

$$\mathcal{L}^A = \lambda^A_{Rec}\mathcal{L}_{Rec} + \lambda^A_{Adv}\mathcal{L}_{Adv} + \lambda^A_{Percep}\mathcal{L}_{Percep}, \tag{17}$$

where $\lambda^A_{Rec} = 1$, $\lambda^A_{Adv} = 10^{-3}$ and $\lambda^A_{Percep} = 0.1$.

Losses for SVBRDF Maps. Let $\mathbf{X}$ be one of $\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S}$, and $\hat{\mathbf{X}}$ be the estimated map. SVBRDF maps are constrained by the above L_1 and a rendering loss [14]. Taking $\mathbf{N}$ as an example, let $\hat{\mathbf{X}}$ be the estimated surface normal map $\hat{\mathbf{N}}$, the rendering loss for $\hat{\mathbf{N}}$ is:

$$\begin{aligned}\mathcal{L}_{Render} = \sum_i^L \|&\mathcal{R}\left((\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S}), (\boldsymbol{l}_i, \boldsymbol{v})\right) - \\ &\mathcal{R}((\hat{\mathbf{N}}, \mathbf{D}, \mathbf{R}, \mathbf{S}, (\boldsymbol{l}_i, \boldsymbol{v}))\|_1,\end{aligned} \tag{18}$$

where $L = 9$ is the number of renderings under different lighting conditions. The total loss of $\mathbf{N}$ is:

$$\mathcal{L}^M = \lambda^M_{Rec}\mathcal{L}_{Rec} + \lambda^M_{Render}\mathcal{L}_{Render}, \tag{19}$$

where $\lambda^M_{Rec} = 1$ and $\lambda^M_{Render} = 0.5$. The training objective for other SVBRDF maps is defined in the same way.

IV. Experiments

In the following, we introduce the experimental settings and compare our method with state-of-the-art approaches. We also provide extensive ablations to validate the effectiveness of the design strategies. More information can be found in the supplemental material.

TABLE I
COMPARISON ON TEST DATASET OF [14] WITH 84 SAMPLES. AVERAGE RMSE OF SVBRDF MAPS ARE PROVIDED. FOLLOWING [17; 47], 9 IMAGES ARE RE-RENDERED FROM SVBRDF MAPS, BASED ON WHICH AVERAGE RMSE AND LPIPS OF RE-RENDERINGS (REN) ARE COMPUTED.

Method	RMSE					LPIPS
	N	D	R	S	Ren	Ren
Des18	0.063	0.034	0.134	0.044	0.157	0.313
Zhou21	0.059	0.030	0.089	0.027	0.130	0.222
Guo21	0.061	0.028	0.083	0.031	0.137	0.241
Gao19	0.066	0.031	0.095	0.034	0.121	0.194
Zhou22	0.062	0.029	0.106	0.036	0.110	0.210
Our	**0.050**	**0.016**	**0.055**	**0.021**	**0.101**	**0.159**

TABLE II
COMPARISON ON TEST DATASET OF [44] WITH 29 SAMPLES. AVERAGE RMSE OF SVBRDF MAPS ARE PROVIDED. FOLLOWING [17; 47], 9 IMAGES ARE RE-RENDERED FROM SVBRDF MAPS, BASED ON WHICH AVERAGE RMSE AND LPIPS OF RE-RENDERINGS (REN) ARE COMPUTED.

Method	RMSE					LPIPS
	N	D	R	S	Ren	Ren
Des18	0.055	0.036	0.202	0.057	0.165	0.353
Zhou21	0.050	0.033	0.115	0.035	0.138	0.227
Guo21	0.052	0.031	0.106	0.036	0.145	0.292
Gao19	0.061	0.029	0.159	0.048	0.122	0.271
Zhou22	0.050	**0.015**	0.076	0.048	**0.086**	0.209
Our	**0.041**	0.018	**0.072**	**0.022**	0.101	**0.180**

A. Experimental Settings

1) Training Dataset: For comparison and ablation study, we use the training dataset provided by [14] which contains 194,068 tuples of $(\mathbf{I}, \mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$, where $(\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$ are artist-created ground-truth SVBRDF maps, and $\mathbf{I}$ is the rendered image based on the $(\mathbf{N}, \mathbf{D}, \mathbf{R}, \mathbf{S})$ according to Eq. 1. Following [17; 47], we choose a half of the dataset for the training by removing similar samples with slightly different viewing and lighting directions. We extend this dataset with mask images $(\mathbf{M}_b, \mathbf{M}_h)$ and auxiliary renderings $(\mathbf{I}_{bf}, \mathbf{I}_{hr}, \mathbf{I}_{hr^2})$, computed by Equations from 4 to 11. The data generation of the mask and aux-rendering images is efficient. We implement the data preparation algorithm in parallel. On an NVIDIA GeForce RTX 3090 GPU, we need 0.029s to render a batch of bumpiness-flatten images $\mathbf{I}_{bf}$ with the batch size of 16. We need 0.164s to render a batch of 16 highlight-removed images $\mathbf{I}_{hr}$ or $\mathbf{I}^2_{hr}$. We need 0.03s to compute a batch of 16 mask images $\mathbf{M}_b$ or $\mathbf{M}_h$. Instead of generating the intermediate images on-the-fly, we choose to prepare them beforehand for saving the training time.

2) Test Datasets: We test our method on both synthetic and real images. The synthetic test datasets include those from [14] containing 84 test samples and from [44] with 29 test samples. We test on real images from [14], [47] and [49].

3) Evaluation Metrics: We adopt two evaluation metrics of Root Mean Squared Error (RMSE) and Learned Perceptual Image Patch Similarity (LPIPS). RMSE evaluates both the estimated SVBRDF maps and the re-rendered images, while LPIPS evaluates the re-rendered images perceptually.

4) Implementation Details: We implement our model using PyTorch, training it on an NVIDIA GeForce RTX 3090 GPU step by step. First, we train the backbone UNets that generate mask images. Second, we fix the backbone UNets and train the GAD-UNets that generate auxiliary images. Finally, we fix both backbone and GAD-UNets, and train the UNets that output SVBRDF maps. All the models are optimized with the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. The learning rate is 2e-4, multiplied by 0.5 after every 60,000 iterations. The backbone UNets and GAD-UNets are trained for 200,000 iterations, while the UNets in the third stage are trained for 300,000 iterations, all with the batch size of 16.

5) Compared Approaches: We compare our method against state-of-the-art flash-based single-image SVBRDF recovery approaches. Specifically, we compare with regression-based approaches [14], [56], [17] and optimization-based approaches [46] and [49]. For [14] and [56], we retrain their models based on the public source code. For the two optimization methods [46] and [49], we directly use their provided pre-trained models. The results of [17] are obtained with the help of the authors of the paper.

B. Results on Synthetic Data

Table I and II show the quantitative comparisons on synthetic data between our method and state-of-the-art approaches. The results in Table I are obtained on the test dataset of [14], and Table II are obtained on the test dataset of [44].

Since ground-truth SVBRDF maps are available, we are able to compute the average RMSE of the estimated results with respect the ground truth. Following [17], we re-render 9 images under different point light sources, and report the average RMSE of the re-rendered images (see "Ren" columns in both tables). Besides, we provide the comparisons in terms of LPIPS of re-rendered images.

In Table I, our method outperforms all the compared approaches in predicting SVBRDF maps. Accordingly, our re-rendered images from the estimated SVBRDF maps are better. In Table. II, our method outperforms regression-based approaches and the optimization-based method [46]. When compared with the latest optimization approach [49], we are better in predicting the normal, roughness and specular maps. In particular, our method is better at predicting the specular component, with an RMSE of 0.022, less than half of 0.048 of [49]. However, our method is slightly worse for the diffuse map (0.018 (our) vs 0.015 ([49])), but note that the method of [49] further optimizes the network parameters at test time, while our method relies purely on regression. For the re-rendered images, our method is perceptually better as measured by the LPIPS metric.

Besides the above numerical comparisons, we also provide qualitative comparisons with previous approaches. The full list of results are provided at weiyintime.github.io/svbrdfdes18 and weiyintime.github.io/svbrdfdes19. In the following, we take several typical examples to perform analysis.

In Figure 8, we provide examples that compare our method against regression-based approaches. For each input image, the ground-truth SVBRDF maps and re-rendered images are presented in the first row. The other rows show the SVBRDF maps estimated by different methods and the corresponding re-renderings. By the first example on the left side of the figure, we show that our method produces better normal maps than

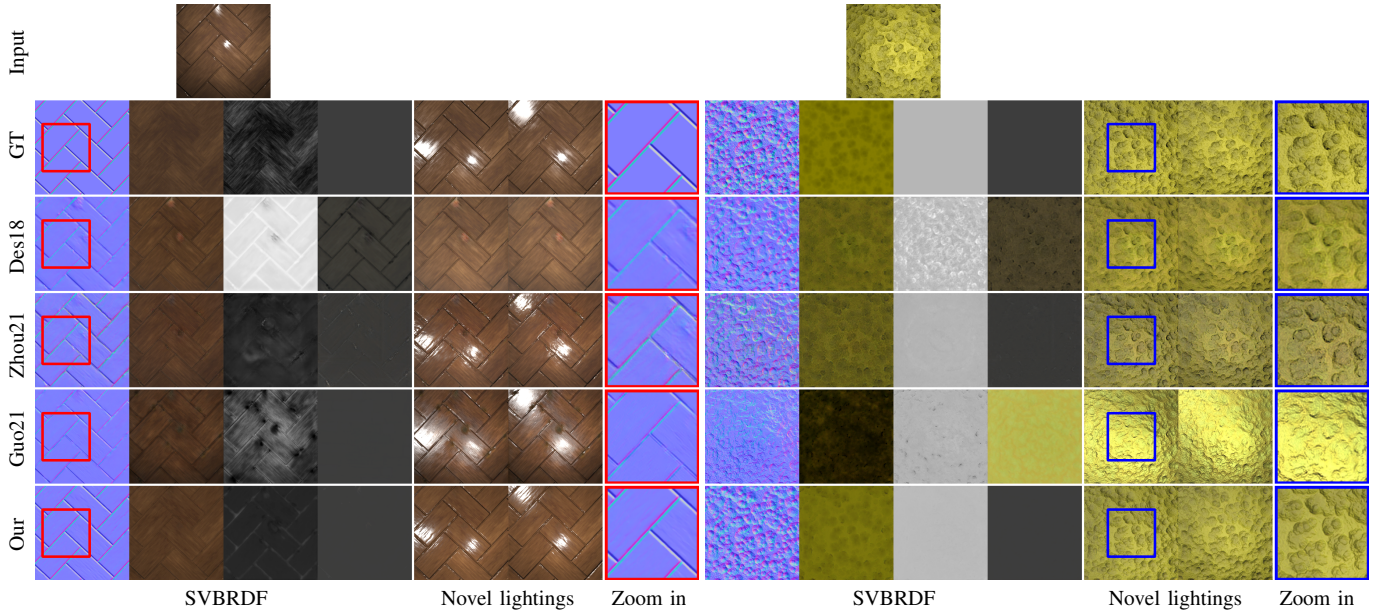

Fig. 8. Comparison with regression-based approaches Des18 [14], Zhou21 [56] and Guo21 [17]. Our method produces more accurate normals (left example), and the color tone of the re-renderings are more similar to the ground truths (right example). See weiyintime.github.io/svbrdfdes18 for more results.

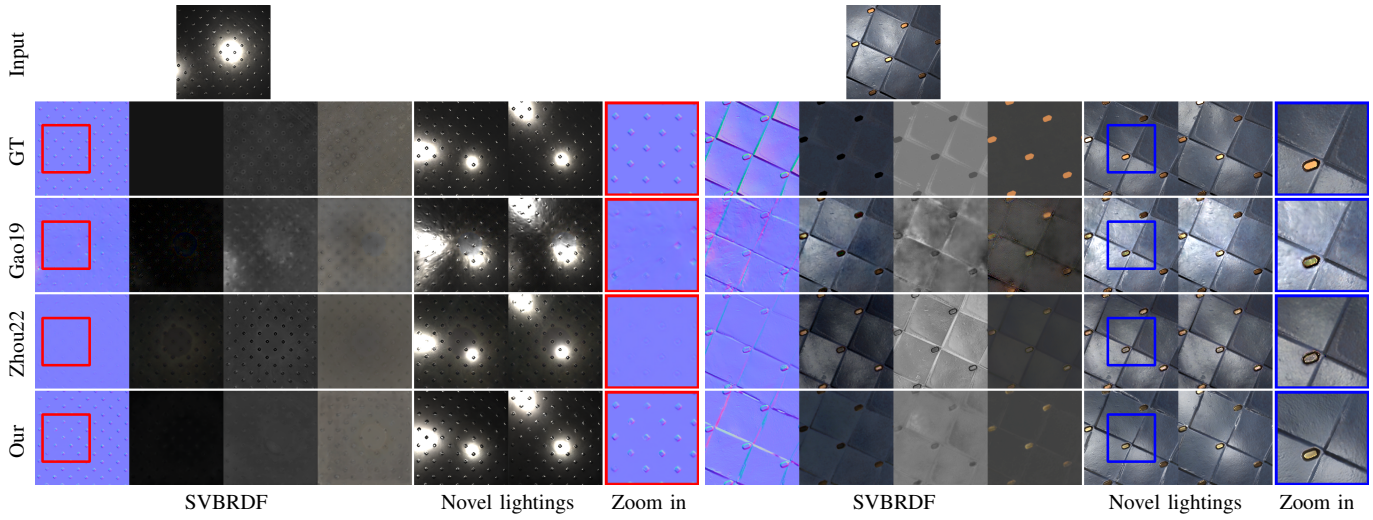

Fig. 9. Comparison with optimization-based approaches Gao19 [46] and Zhou22 [49]. Our method produces more accurate normals (left example), and the metal block shows shinier luster (right example). See weiyintime.github.io/svbrdfdes19 for more results.

the other approaches, as can be seen in the "zoom-in" images. Our estimated normals exhibit clear brick boundaries due to the introduction of the bumpiness-flattened auxiliary image $\mathbf{I}_{bf}$, while the normals of other methods are blurry. On the right side we show an example for which our method outputs perfect SVBRDF maps, while other approaches suffer from artifacts in the diffuse map (Guo21 [17]) or the roughness map (Des18 [14]). The re-renderings of our method resemble the ground truths more, as illustrated in the "zoom-in" images on the right side of the figure.

In Figure 9, we show examples comparing against optimization-based approaches. The two examples further validate that our method outputs better normal maps. For the example on the left side, the method of [49] fails to recognize the tiny protrusions of the metal material, and the method of [46] incorrectly treats the protrusions as concave holes. In contrast, our method recovers most of the normals correctly. For the example on the right side, please pay attention to the specular map containing scattered yellow metallic blocks, which in our experience are challenging to regress. Thanks to the introduction of the pair of $(\mathbf{I}_{hr}, \mathbf{I}_{hr^2})$, our method disentangles the metallic material from the gray background better than the two optimization approaches. As a result, in our re-rendered images, the metallic block shows shinier metallic luster, as shown in the "zoom-in" images.

C. Results on Real Data

In Figure 10, we compare with regression-based approaches on real images qualitatively (more results can be found at weiyintime.github.io/svbrdfreal). For the two examples, since

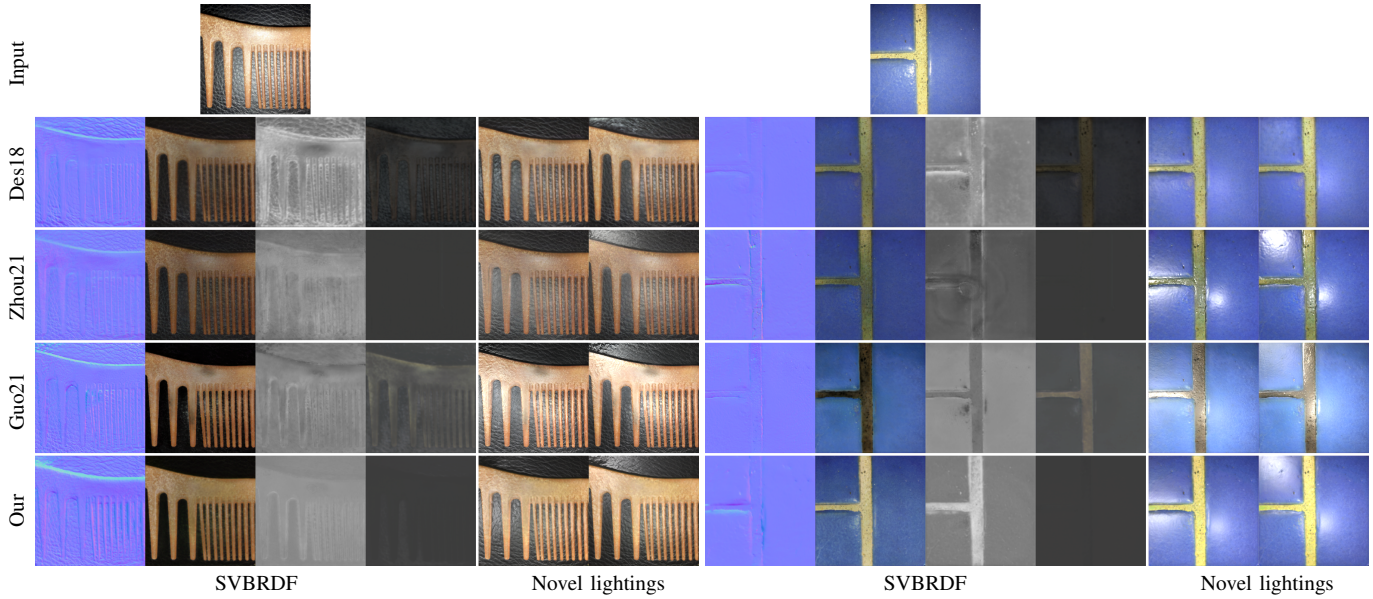

Fig. 10. Comparison with regression-based approaches Des18 [14], Zhou21 [56] and Guo21 [17] qualitatively on real images. Left: previous approaches suffer from highlight artifacts, while our method does not. Right: our roughness map is more reasonable and accordingly the re-renderings are more vivid. See weiyintime.github.io/svbrdfreal for more results.

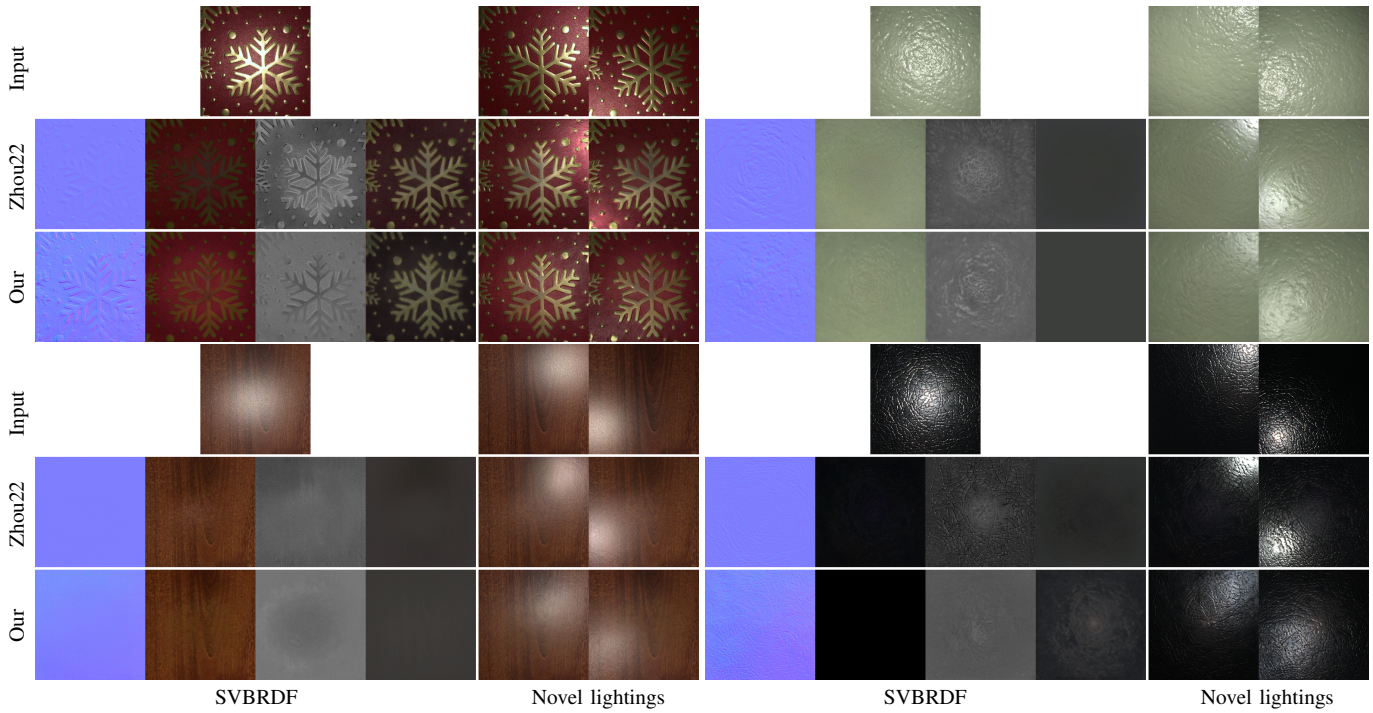

Fig. 11. Comparison with Zhou22 [49] on real images. Consistent with synthetic results, our method recovers prominent normals for real inputs (see the first example). In the re-renderings, our produced highlights own slightly weaker intensity (we provide more analysis in the supplemental material), but the shape of our highlights resembles that of ground truths more.

there are no ground-truth SVBRDF maps, expertise is required to compare the estimated results of different approaches. Overall in both examples, our method produces more vivid normal maps and re-rendered images. Please see the diffuse maps of the left example. We observe that the highlight remains in the diffuse maps estimated by [14] and [56]. The method of [17] attempts to remove the highlight, but fails, as shown by the shadows in the original highlight region. In contrast, our method avoids the above artifacts in the diffuse map, and accordingly avoids artifacts in the re-renderings under novel lightings. For the second example, the input image exhibits highlight in the blue background but not in the yellow area, which means the yellow region should have higher roughness. The roughness map estimated by our method is consistent with this finding, while other roughness maps are not.

We also conduct comparisons on real images against

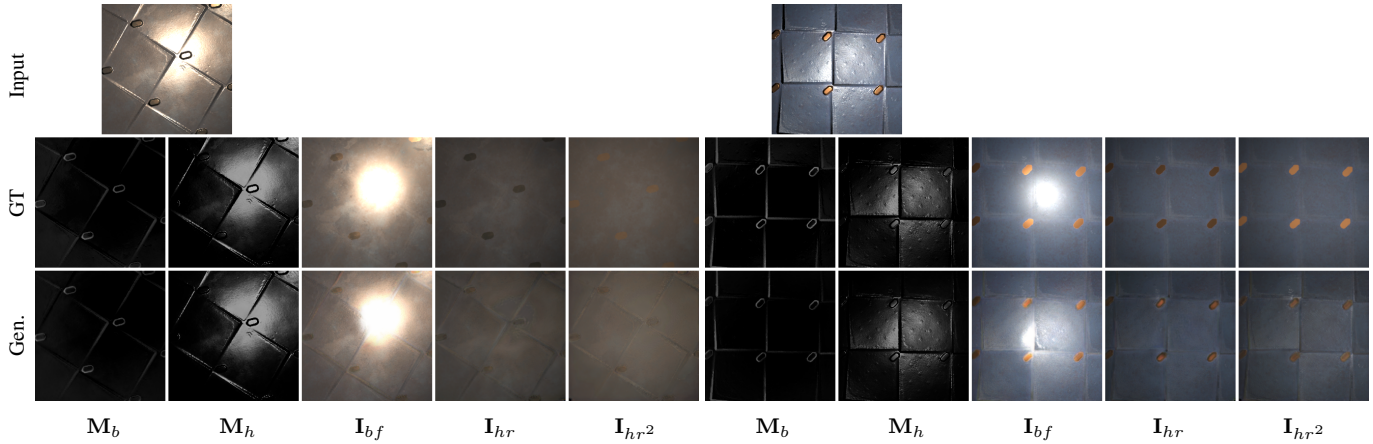

Fig. 12. Intermediate mask and auxiliary images generated by our method. The two examples are very challenging, but our method can still output visually similar results to the ground truths. Please refer to the supplemental material for more examples.

TABLE III
COMPARISON WITH OPTIMIZATION-BASED APPROACHES ON THE REAL TEST IMAGES OF [47] (CONTAINING 33 SCENES) AND [49] (76 SCENES) QUANTITATIVELY. BOLD NUMBERS ARE THE BEST, AND UNDERLINED NUMBERS ARE THE SECOND BEST.

Method	Dataset of [47]		Dataset of [49]	
	LPIPS	RMSE	LPIPS	RMSE
Gao19	0.361	0.158	0.290	0.110
Zhou22	<u>0.286</u>	**0.133**	**0.216**	**0.093**
Our	**0.282**	<u>0.143</u>	<u>0.222</u>	<u>0.098</u>

optimization-based approaches. Guo et al. [47] and Zhou et al. [49] collect test datasets consisting of 33 and 76 scenes, respectively. For each scene, they capture a set of 9 images illuminated with known lightings. Using one of them as the input image to estimate SVBRDF maps, the re-rendered images under the other 8 lightings can be compared with the ground truths to output numerical values measuring the effectiveness of an approach. To compare with the optimization-based approaches fairly, we design a scheme that optimizes our results too. We use the same loss function as [49] used in the test-time optimizations, and the same optimizer of Adam with parameters of $(\beta_1 = 0.5, \beta_2 = 0.5)$. Starting from the initial SVBRDF maps by our model, we further optimize the UNets in the third stage of our method 100 times (please see the supplemental material for the influence of the optimization times) using the learning rate of 1e-5. The numerical results are shown in Table III. Figure 11 shows some examples of [49] and our method. As can be seen, our method is better than [46], and is on par with [49]. However, although [49] is better than our method numerically, it has the shortcoming that the lighting position of the test image must be known in advance or the input image must be captured with centered lighting, while our method is not subject to this limitation.

D. Intermediate Results

In Figure 12, we show examples of the mask and auxiliary images generated by our method. As can be seen, the generated mask images (including $\mathbf{M}_b$ and $\mathbf{M}_h$) look very similar to the corresponding ground-truth mask images, validating the effectiveness of the backbone UNets incorporating EfficientNet-B3 to extract basic features from the test images. Our method also generates satisfactory $\mathbf{I}_{bf}$, which effectively reduces the bumpiness effect in the input images. For the other two auxiliary images $\mathbf{I}_{hr}$ and $\mathbf{I}_{hr^2}$, note that they are more different from the input image than $\mathbf{I}_{bf}$, thus we are not able to produce results very similar to the ground truths, but our method still outputs results that successfully get rid of highlights and remove bumpiness as much as possible. In fact, the two examples are very challenging to extract intermediate images. Due to space limit, please refer to the supplemental material for more intermediate results of both synthetic and real images with more discussions.

TABLE IV
STEP-BY-STEP ERROR ANALYSIS. ① GIVES THE RESULTS OF OUR FINAL MODEL. ② IS SIMILAR TO ABLATION ① EXCEPT THAT GT MASK IMAGES ARE USED. ③ DIRECTLY MAPS GT AUX-RENDERINGS TO SVBRDF MAPS, WHICH GIVES THE UPPER-BOUND ACCURACY OF OUR METHOD. ④ ONLY USES AUX-RENDERINGS AS THE INTERMEDIATE TARGETS, WHILE ABANDONING THE MASK IMAGES. ⑤ DIRECTLY MAPS AN INPUT IMAGE TO SVBRDF MAPS.

Ablation Scheme	N	D	R	S
①Input→Mask→Aux→SVBRDF (Our)	0.050	0.016	0.055	0.021
②Input→GT Mask→Aux→SVBRDF	0.044	0.012	0.046	0.018
③GT Aux→SVBRDF (Upper-bound accuracy)	0.035	0.010	0.032	0.015
④Input→Aux→SVBRDF	0.052	0.024	0.071	0.026
⑤Input→SVBRDF	0.066	0.035	0.154	0.045

E. Ablation Studies

We conduct ablation studies to validate the necessity of every component of the proposed method. All the following ablation results are obtained on the test dataset of [14], measured by RMSE.

1) Step-by-Step Error Analysis.: Our method contains multiple steps containing networks trained step-by-step. Each phase may produce errors and the errors may propagate to the final predictions. In this section, we analyze these errors step-by-step by a group of 5 ablation experiments in Table IV. Ablation ① gives results of our final model which first infers mask images, then the aux-renderings, and finally the SVBRDF maps. Ablation ② is similar to ①, except that GT

TABLE V
ABLATION ON TRAINING STRATEGIES. "W/O INTERMEDIATE LOSSES" MEANS NOT APPLYING LOSSES TO BOTH MASK AND AUX-RENDERINGS.

Ablation Scheme	N	D	R	S
End-to-end w/o intermediate losses	0.057	0.029	0.091	0.033
End-to-end w/ intermediate losses	0.052	0.022	0.073	0.024
Our final model	0.050	0.016	0.055	0.021

mask images are adopted. From the comparison between ① and ②, we see that the errors in masks reduce the accuracy of (N,D,R,S) by (0.006,0.004,0.009,0.003) respectively. In ③, we directly map GT aux-renderings to SVBRDF maps. In this case, there is no error in the intermediate images by which we can theoretically obtain the upper-bound accuracy of our method which is (0.035,0.010,0.032,0.015) respectively.

Ablation ② and ③ increase the accuracy of the intermediate images, thus obtaining better final predictions. On the contrary in ablation ④, we decrease the accuracy of the aux-renderings by not using the mask images. The accuracy of the final predictions decreases too, demonstrating the usefulness of the mask images. In ablation ⑤, we directly map input images to SVBRDF maps without any intermediate target, producing the worst results among the 5 ablations.

2) Ablations on Training Strategies and Intermediate Losses: We train our final model stage-by-stage. Another way is to train the model end-to-end. If end-to-end, we can either apply losses to the intermediate results or not. In Table V, we perform ablations on different training strategies, and find that end-to-end training is more likely to get stuck in local minima, while stage-by-stage training is easier to control, converging faster and outputting better results.

3) Ablations on Inputting Alternative Combinations of Intermediate Results for SVBRDF Inference: In Table VI, we test alternative input combinations for estimating SVBRDF maps. As shown on the left side, the last line shows our finally adopted input which is composed of $\mathbf{I}$ and $\mathbf{I}-\mathbf{I}_{bf}$ for inferring $\mathbf{N}$. We test only using $\mathbf{I}$ as input which yields large accuracy drop. We test ($\mathbf{I}$,$\mathbf{I}-\mathbf{I}_{bf}$,$\mathbf{M}_b$) as input and obtain similar result as inputting ($\mathbf{I}$,$\mathbf{I}-\mathbf{I}_{bf}$), possibly because $\mathbf{I}-\mathbf{I}_{bf}$ already conveys information about $\mathbf{M}_b$. On the right side, we test different kinds of input combinations for inferring the diffuse map. Similarly, using $\mathbf{I}$ as input yields worst result. We then experiment with additionally inputting either $\mathbf{I}_{hr}$ or $\mathbf{I}_{hr^2}$ besides $\mathbf{I}$, which are also inferior to using all the three elements as input. Finally, we test ($\mathbf{I}$,$\mathbf{I}_{bf}$,$\mathbf{I}_{hr}$,$\mathbf{I}_{hr^2}$), which is inferior too, possibly due to the highlights in $\mathbf{I}_{bf}$.

4) Ablations on UNets Used in Each Stage: We use backbone UNets in the first stage, aiming at relying on the large pre-trained backbone model for identifying mask edges. We employ gated deformable UNets in the second stage, as it can better remove and inpaint highlights. We use the original form of UNet in the third stage because this simple network can fulfill the tasks in this stage as demonstrated in Section IV-E1.

We conduct ablations to validate our choices of UNets in Table VII. As seen on the top part of the table, we use different kinds of UNets to infer the bumpiness mask $\mathbf{M}_b$ from the input image $\mathbf{I}$. It is the backbone UNet outputs the lowest RMSE. From the part at the bottom of the table, we see that

TABLE VI
ABLATION ON ALTERNATIVE INPUT COMBINATIONS FOR ESTIMATING SVBRDF MAPS. ROWS WITH GRAY BACKGROUND PROVIDE THE ADOPTED SCHEME.

Input	Target	RMSE	Input	Target	RMSE
$\mathbf{I}$	$\mathbf{N}$	0.066	$\mathbf{I}$	$\mathbf{D}$	0.035
($\mathbf{I}$,$\mathbf{I}$-$\mathbf{I}_{bf}$,$\mathbf{M}_b$)	$\mathbf{N}$	0.051	($\mathbf{I}$,$\mathbf{I}_{bf}$,$\mathbf{I}_{hr}$,$\mathbf{I}_{hr^2}$)	$\mathbf{D}$	0.019
($\mathbf{I}$,$\mathbf{I}$-$\mathbf{I}_{bf}$)	$\mathbf{N}$	0.050	($\mathbf{I}$,$\mathbf{I}_{hr}$)	$\mathbf{D}$	0.025
			($\mathbf{I}$,$\mathbf{I}_{hr^2}$)	$\mathbf{D}$	0.028
			($\mathbf{I}$,$\mathbf{I}_{hr}$,$\mathbf{I}_{hr^2}$)	$\mathbf{D}$	0.016

TABLE VII
ABLATION ON APPLYING OTHER UNETS FOR INFERRING MASK IMAGES AND AUX-RENDERINGS. "ORI. UNET": ORIGINAL UNET, "GAD-UNET": GATED DEFORMABLE UNET, "BB-UNET": BACKBONE UNET, AND "HA-UNET": THE HIGHLIGHT-AWARE UNET PROPOSED IN [17].

Task	Model	RMSE
$\mathbf{I} \rightarrow \mathbf{M}_b$	Ori. UNet	0.030
$\mathbf{I} \rightarrow \mathbf{M}_b$	GAD-UNet	0.028
$\mathbf{I} \rightarrow \mathbf{M}_b$	BB-UNet	0.023
$(\mathbf{I}, \mathbf{M}_b, \mathbf{M}_h) -> \mathbf{I}_{hr}$	Ori. UNet	0.033
$(\mathbf{I}, \mathbf{M}_b, \mathbf{M}_h) -> \mathbf{I}_{hr}$	BB-UNet	0.031
$(\mathbf{I}, \mathbf{M}_b, \mathbf{M}_h) -> \mathbf{I}_{hr}$	HA-UNet	0.029
$(\mathbf{I}, \mathbf{M}_b, \mathbf{M}_h) -> \mathbf{I}_{hr}$	GAD-UNet	0.025

the gated deformable UNet generates better $\mathbf{I}_{hr}$ than other kinds of UNets including the HA-UNet proposed in [17].

5) $\mathbf{I}_{hr^2}$ vs. $\mathbf{D}+\mathbf{S}$: As mentioned above, $\mathbf{I}_{hr^2}$ is potentially equal to the sum of diffuse and specular (i.e., $\mathbf{D}+\mathbf{S}$), as they are both mainly determined by the diffuse and specular components. However, the formation of $\mathbf{I}_{hr^2}$ is essentially influenced by the lighting conditions, and since $\mathbf{I}_{hr^2}$ share similar lighting conditions as the input image $\mathbf{I}$, we expect it is easier to infer $\mathbf{I}_{hr^2}$ than $\mathbf{D}+\mathbf{S}$ from $\mathbf{I}$. To prove this, we conduct the following experiment: we generate $\mathbf{D}+\mathbf{S}$ using the same way generating $\mathbf{I}_{hr^2}$. After testing, we obtain an average RMSE of 0.054 for $\mathbf{D}+\mathbf{S}$, and 0.030 for $\mathbf{I}_{hr^2}$, which validate our expectation and therefore we prefer using $\mathbf{I}_{hr^2}$ to $\mathbf{D}+\mathbf{S}$.

F. Discussions and Limitations

1) Model Size: In our model, the size of a backbone UNet is 7.4M, the size of a gated deformable UNet is 25M, and the size of a UNet in the third stage is 15M. The total size of our model is around 135M. In comparison, the model size of [14] is 81M. Our model size is not much larger than that of [14] even we have multiple UNets whereas [14] has only one. This is because we use a small base channel size of 48 in our UNets, while that of [14] is 64.

2) Training and Running Time: Training a backbone UNet costs around 1.5 days. Training a UNet in the second or third stages costs about 2.5 to 3 days. At test phase, we take about 0.107 seconds to output four SVBRDF maps. The running time for Des18 [14] is 0.086s, Zhou21 [56] is 0.065s, Guo21 [17] is 0.038s, Gao19 [46] is 57.298s, and Zhou22 [49] is around 5.26s.

3) Extension to BRDF models Other Than Cook-Torrance: We believe the philosophy that guides us to design the three auxiliary renderings can be also used to design auxiliary renderings for other parametric material models like BSDF/BTDF. For example, we can change the refractive

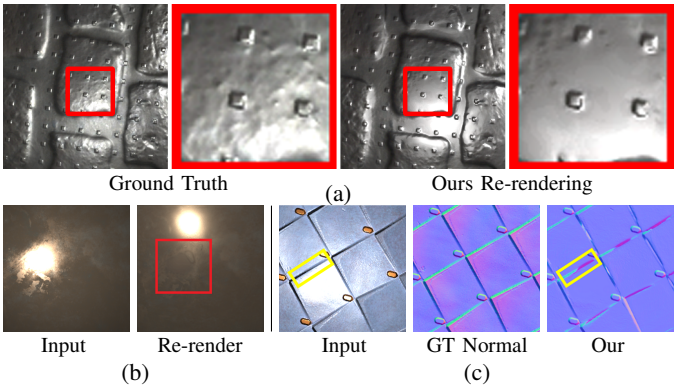

Fig. 13. (a) Producing smoother results with fewer details than GT. (b) Cannot perfectly inpaint highlight when it is strong and large in size. (c) Sometimes predicting contrary normal directions for different parts of the same objects.

index to generate different images, from which we can infer the index of refraction. But with no doubt, finding effective designs of intermediate maps requires a lot of trial and error.

4) Limitations and Future Works: A problem with our method is that it is composed of many stages, and it is better to train our model stage-by-stage than end-to-end, as demonstrated by experiments in Table V. This makes the training of our model inconvenient and also increases the training time. In the future, we would like to investigate more about why end-to-end training is inferior in our situation.

We find that compared with the ground-truth re-rendered images, our re-renderings are sometimes smoother and lose details (see Figure 13 (a)). We ascribe this to the use of the $L1$ loss in Eq. 14. Due to the higher regression power of our method, the $L1$ loss may force our network to learn SVBRDF maps that look like ground truths on average but fail to regress local details. We have ever applied GAN losses to SVBRDF maps to alleviate this problem which however hurt the numerical accuracy of the maps. How to solve this problem deserves future research.

Besides, when the highlight is too strong and large in size (see Figure 13 (b)), our method may fail to perfectly inpaint the highlight region. This is a problem in the field of large-hole inpainting, and may be solved if drawing on the latest inspiration in that field.

We also find that our method cannot predict completely correct normals for the example (and some similar ones (see the full list of results on website)) in Figure 13 (c). It seems our method is susceptible to local features. For example, for the area in the yellow box, the darker half (see the input image) is predicted as green normals, while the lighter half is predicted as red normals, unable to treat the two parts as a single object. Maybe, it is necessary to introduce a module into our model that can model long-range relationships between features.

For more discussions about limitations and future research directions, and results of high-resolution images can be found in the supplemental material.

V. Conclusion

This paper presents a novel three-stage SVBRDF estimation method from a single image lit by a flash. We contributed to the community a new paradigm for solving this problem: inferring intermediate targets from the input image at first, and then inferring the SVBRDF maps from the intermediate targets. We proposed three auxiliary renderings and two mask images. The pipeline preparing these intermediate and auxiliary images is novel, achieved by modifying the original normal and roughness maps during the forward rendering process (for the auxiliary renderings), or even modifying the rendering function itself (for the mask images). Based on the five intermediate images, we proposed a single-image SVBRDF estimation framework composed of three well-designed UNet-based networks to generate mask, auxiliary and the final SVBRDF images, respectively. We have conducted extensive experiments on both synthetic and real datasets to validate the effectiveness of the proposed method.

Acknowledgments

This work was supported by the Natural Science Foundation of China (62072191,U21A20520,62325204), the National Key Research and Development Program of China (2022YFE0112200), the Key-Area Research and Development Program of Guangzhou City (202206030009), and the Guangdong Basic and Applied Basic Research Fund (2023A1515030002,2024A1515011995).

References

[1] R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," *ACM Transactions on Graphics (TOG)*, vol. 1, no. 1, pp. 7–24, 1982.

[2] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM TOG*, vol. 18, no. 1, pp. 1–34, 1999.

[3] K. J. Dana and J. Wang, "Device for convenient measurement of spatially varying bidirectional reflectance," *JOSA A*, vol. 21, no. 1, pp. 1–12, 2004.

[4] J. Lawrence, A. Ben-Artzi, C. DeCoro, W. Matusik, H. Pfister, R. Ramamoorthi, and S. Rusinkiewicz, "Inverse shade trees for non-parametric material representation and editing," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 735–745, 2006.

[5] M. Holroyd, J. Lawrence, and T. Zickler, "A coaxial optical scanner for synchronous acquisition of 3d geometry and surface reflectance," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–12, 2010.

[6] Y. Dong, J. Wang, X. Tong, J. Snyder, Y. Lan, M. Ben-Ezra, and B. Guo, "Manifold bootstrapping for svbrdf capture," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–10, 2010.

[7] M. Aittala, T. Weyrich, and J. Lehtinen, "Practical svbrdf capture in the frequency domain." *ACM Trans. Graph.*, vol. 32, no. 4, pp. 110–1, 2013.

[8] M. Aittala, T. Weyrich, J. Lehtinen *et al.*, "Two-shot svbrdf capture for stationary materials." *ACM Trans. Graph.*, vol. 34, no. 4, pp. 110–1, 2015.

[9] K. Kang, Z. Chen, J. Wang, K. Zhou, and H. Wu, "Efficient reflectance capture using an autoencoder." *ACM Trans. Graph.*, vol. 37, no. 4, pp. 127–1, 2018.

[10] Z. Zhou, G. Chen, Y. Dong, D. Wipf, Y. Yu, J. Snyder, and X. Tong, "Sparse-as-possible svbrdf acquisition," *ACM TOG*, vol. 35, no. 6, pp. 1–12, 2016.

[11] Z. Hui, K. Sunkavalli, J.-Y. Lee, S. Hadap, J. Wang, and A. C. Sankaranarayanan, "Reflectance capture using univariate sampling of brdfs," in *Proceedings of ICCV*, 2017, pp. 5362–5370.

[12] P. Ren, J. Wang, J. Snyder, X. Tong, and B. Guo, "Pocket reflectometry," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, pp. 1–10, 2011.

[13] 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.

[14] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, "Single-image svbrdf capture with a rendering-aware deep network," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–15, 2018.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[16] Z. Li, K. Sunkavalli, and M. Chandraker, "Materials for masses: Svbrdf acquisition with a single mobile phone image," in *Proceedings of ECCV*, 2018, pp. 72–87.

[17] J. Guo, S. Lai, C. Tao, Y. Cai, L. Wang, Y. Guo, and L.-Q. Yan, "Highlight-aware two-stream network for single-image svbrdf acquisition," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[18] R. Martin, A. Roullier, R. Rouffet, A. Kaiser, and T. Boubekeur, "Materia: Single image high-resolution material capture in the wild," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 163–177.

[19] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*. PMLR, 2019, pp. 6105–6114.

[20] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 511–518.

[21] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces." *Rendering techniques*, vol. 2007, p. 18th, 2007.

[22] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometrical considerations and nomenclature for reflectance*. USA: Jones and Bartlett Publishers, Inc., 1992, p. 94–145.

[23] D. Guarnera, G. C. Guarnera, A. Ghosh, C. Denk, and M. Glencross, "Brdf representation and acquisition," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 625–650.

[24] Y. Francken, T. Cuypers, T. Mertens, J. Gielis, and P. Bekaert, "High quality mesostructure acquisition using specularities," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–7.

[25] C.-P. Wang, N. Snavely, and S. Marschner, "Estimating dual-scale properties of glossy surfaces from step-edge lighting," in *Proceedings of the 2011 SIGGRAPH Asia Conference*, 2011, pp. 1–12.

[26] G. C. Guarnera, P. Peers, P. Debevec, and A. Ghosh, "Estimating surface normals from spherical stokes reflectance fields," in *European Conference on Computer Vision*. Springer, 2012, pp. 340–349.

[27] G. Chen, Y. Dong, P. Peers, J. Zhang, and X. Tong, "Reflectance scanning: Estimating shading frame and brdf with generalized linear light sources," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014.

[28] L.-P. Asselin, D. Laurendeau, and J.-F. Lalonde, "Deep svbrdf estimation on real materials," in *3DV*. IEEE, 2020, pp. 1157–1166.

[29] S.-H. Baek, D. S. Jeon, X. Tong, and M. H. Kim, "Simultaneous acquisition of polarimetric svbrdf and normals." *ACM TOG*, vol. 37, no. 6, pp. 268–1, 2018.

[30] 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, 2019.

[31] J. Yu, Z. Xu, M. Mannino, H. W. Jensen, and R. Ramamoorthi, "Sparse Sampling for Image-Based SVBRDF Acquisition," in *Workshop on Material Appearance Modeling*, R. Klein and H. Rushmeier, Eds. The Eurographics Association, 2016.

[32] 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.

[33] J. Riviere, P. Peers, and A. Ghosh, "Mobile surface reflectometry," in *Computer Graphics Forum*, vol. 35, no. 1. Wiley Online Library, 2016, pp. 191–202.

[34] 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.

[35] H. Ha, S.-H. Baek, G. Nam, and M. H. Kim, "Progressive acquisition of svbrdf and shape in motion," in *Computer Graphics Forum*, vol. 39, no. 6. Wiley Online Library, 2020, pp. 480–495.

[36] M. Boss, V. Jampani, K. Kim, H. Lensch, and J. Kautz, "Two-shot spatially-varying brdf and shape estimation," in *Proceedings of CVPR*, 2020, pp. 3982–3991.

[37] R. Pacanowski, O. S. Celis, C. Schlick, X. Granier, P. Poulin, and A. Cuyt, "Rational brdf," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 11, pp. 1824–1835, 2012.

[38] G. Ward, M. Kurt, and N. Bonneel, "Reducing anisotropic bsdf measurement to common practice." in *Material Appearance Modeling*, 2014, pp. 5–8.

[39] M. Aittala, T. Aila, and J. Lehtinen, "Reflectance modeling by neural texture synthesis," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–13, 2016.

[40] X. Li, Y. Dong, P. Peers, and X. Tong, "Modeling surface appearance from a single photograph using self-augmented convolutional neural networks," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, 2017.

[41] W. Ye, X. Li, Y. Dong, P. Peers, and X. Tong, "Single image surface appearance modeling with self-augmented cnns and inexact supervision," in *Computer Graphics Forum*, vol. 37, no. 7. Wiley Online Library, 2018, pp. 201–211.

[42] P. Henzler, V. Deschaintre, N. J. Mitra, and T. Ritschel, "Generative modelling of brdf textures from flash images," *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021.

[43] L. Wang, L. Zhang, F. Gao, and J. Zhang, "Deepbasis: Hand-held single-image svbrdf capture via two-level basis material model," in *SIGGRAPH Asia 2023 Conference Papers*, 2023, pp. 1–11.

[44] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, "Flexible svbrdf capture with a multi-image deep network," in *Computer Graphics Forum*, vol. 38, no. 4. Wiley Online Library, 2019, pp. 1–13.

[45] V. Deschaintre, G. Drettakis, and A. Bousseau, "Guided fine-tuning for large-scale material transfer," in *Computer Graphics Forum*, vol. 39, no. 4. Wiley Online Library, 2020, pp. 91–105.

[46] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong, "Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–15, 2019.

[47] Y. Guo, C. Smith, M. Hašan, K. Sunkavalli, and S. Zhao, "Materialgan: reflectance capture using a generative svbrdf model," *ACM Trans. Graph.*, vol. 39, no. 6, nov 2020.

[48] T. Ono, H. Kubo, K. Tanaka, T. Funatomi, and Y. Mukaigawa, "Practical brdf reconstruction using reliable geometric regions from multi-view stereo," *Computational Visual Media*, vol. 5, no. 4, pp. 325–336, 2019.

[49] X. Zhou and N. K. Kalantari, "Look-ahead training with learned reflectance loss for single-image svbrdf estimation," *ACM TOG*, vol. 41, no. 6, pp. 1–12, 2022.

[50] M. Fischer and T. Ritschel, "Metappearance: Meta-learning for visual appearance reproduction," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–13, 2022.

[51] C. Fan, Y. Lin, and A. Ghosh, "Deep shape and svbrdf estimation using smartphone multi-lens imaging," in *Computer Graphics Forum*, vol. 42, no. 7. Wiley Online Library, 2023, p. e14972.

[52] X. Zhou, M. Hasan, V. Deschaintre, P. Guerrero, Y. Hold-Geoffroy, K. Sunkavalli, and N. K. Kalantari, "Photomat: A material generator learned from single flash photos," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–11.

[53] S. Sartor and P. Peers, "Matfusion: a generative diffusion model for svbrdf capture," in *SIGGRAPH Asia 2023 Conference Papers*, 2023, pp. 1–10.

[54] G. Vecchio, R. Martin, A. Roullier, A. Kaiser, R. Rouffet, V. Deschaintre, and T. Boubekeur, "Controlmat: A controlled generative approach to material capture," *arXiv preprint arXiv:2309.01700*, 2023.

[55] G. Vecchio, S. Palazzo, and C. Spampinato, "Surfacenet: Adversarial svbrdf estimation from a single image," in *Proceedings of ICCV*, 2021, pp. 12 840–12 848.

[56] X. Zhou and N. K. Kalantari, "Adversarial single-image svbrdf estimation with hybrid training," in *Computer Graphics Forum*, vol. 40, no. 2. Wiley Online Library, 2021, pp. 315–325.

[57] Y. Zhao, B. Wang, Y. Xu, Z. Zeng, L. Wang, and N. Holzschuch, "Joint svbrdf recovery and synthesis from a single image using an unsupervised generative adversarial network." in *EGSR (DL)*, 2020, pp. 53–66.

[58] T. Wen, B. Wang, L. Zhang, J. Guo, and N. Holzschuch, "Svbrdf recovery from a single image with highlights using a pre-trained generative adversarial network," in *Computer Graphics Forum*. Wiley Online Library, 2022.

[59] W. Ye, Y. Dong, P. Peers, and B. Guo, "Deep reflectance scanning: Recovering spatially-varying material appearance from a flash-lit video sequence," in *Computer Graphics Forum*, vol. 40, no. 6. Wiley Online Library, 2021, pp. 409–427.

[60] Y. Hu, M. Hašan, P. Guerrero, H. Rushmeier, and V. Deschaintre, "Controlling material appearance by examples," in *Computer Graphics Forum*, vol. 41, no. 4. Wiley Online Library, 2022, pp. 117–128.

[61] X. Zhou, M. Hasan, V. Deschaintre, P. Guerrero, K. Sunkavalli, and N. K. Kalantari, "Tilegen: Tileable, controllable material generation and capture," in *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–9.

[62] X. Zhou, M. Hašan, V. Deschaintre, P. Guerrero, K. Sunkavalli, and N. K. Kalantari, "A semi-procedural convolutional material prior," in *Computer Graphics Forum*, vol. 42, no. 6. Wiley Online Library, 2023, p. e14781.

[63] J. Guo, S. Lai, Q. Tu, C. Tao, C. Zou, and Y. Guo, "Ultra-high resolution svbrdf recovery from a single image," *ACM Transactions on Graphics*, 2023.

[64] L. Zhang, F. Gao, L. Wang, M. Yu, J. Cheng, and J. Zhang, "Deep svbrdf estimation from single image under learned planar lighting," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–11.

[65] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Learning to reconstruct shape and spatially-varying reflectance from a single image," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–11, 2018.

[66] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2475–2484.

[67] M. Tetzlaff, "High-fidelity specular svbrdf acquisition from flash photographs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 4, pp. 1885–1896, 2024.

[68] Z. Huang, K. Hu, and X. Wang, "M2-net: Multi-stages specular highlight detection and removal in multi-scenes," *arXiv:2207.09965*, 2022.

[69] Y. Zhu, J. Huang, X. Fu, F. Zhao, Q. Sun, and Z.-J. Zha, "Bijective mapping network for shadow removal," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5627–5636.

[70] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[71] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*. PMLR, 2017, pp. 214–223.

Yongwei Nie (Member, IEEE) received the BSc and PhD degrees from the Computer School of Wuhan University, in 2009 and 2015, respectively. Currently, he is an associate professor with the School of Computer Science and Engineering, South China University of Technology. His research interests include image and video editing, and computational photography.

Jiaqi Yu received the BS and MS degrees from the South China University of Technology, in 2021 and 2023, respectively. He is currently a graphics engineer with NetEase Games, Guangzhou. His research interests include computer graphics and computer vision.

Chengjiang Long is currently a Research Scientist at Meta Reality Labs. Prior joining Meta, he worked as a Principal Scientist/Tech Leader in JD Tech R&D Center at Silicon Valley (a part of JD.COM) from June 2020 to Dec 2021, and worked as a Computer Vision Researcher/Senior R&D Engineer at Kitware from February 2016 to April 2020. He received the M.S. degree in Computer Science from Wuhan University in 2011 and a B.S degree in Computer Science and Technology from Wuhan University in 2009. He got his Ph.D. degree in Computer Science from Stevens Institute of Technology in 2015. His research interests involve various areas of Computer Vision, Computer Graphics, Multimedia, Machine Learning, and Artificial Intelligence. He is a member of IEEE and AAAI.

Qing Zhang is an associate professor with the School of Computer Science and Engineering, Sun Yat-sen University. His research interests include computational photography, computer vision, and computer graphics.

Guiqing Li received the BS degree from the University of Science and Technology of China, in 1987, the MS degree from Nankai University, in 1990, and the PhD degree from the Institute of Computing Technology, CAS, in 2001. He is now a professor with the School of Computer Science and Engineering, South China University of Technology (SCUT). Before joining SCUT, he worked as a postdoctoral researcher in State Key Lab of CAD & CG, Zhejiang University (2001 2003). He was a lecturer and then Associate Professor with the School of Computer and Information Engineering, Guangxi University (1994 1998), China. He became a teacher of the department of mathematics, Guangxi College of Education in 1990. From 2002 to 2008, he visited the City University of Hong Kong several times for short-term research. He has published more than 100 papers. His research interests include CAGD, digital geometry processing, 3D animation, motion object tracking, and image/video editing.

Hongmin Cai (Senior member, IEEE) is a Professor at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He received the B.S. and M.S. degrees in mathematics from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree in applied mathematics from Hong Kong University in 2007. His areas of research interests include biomedical image processing and omics data integration.