Inverse Rendering of Translucent Objects using Physical and Neural Renderers

Chenhao Li, Trung Thanh Ngo, Hajime Nagahara

Institute for Datability Science, Osaka University

{lch@is.ids, nagahara@ids, trung@am.sanken}.osaka-u.ac.jp

Abstract

In this work, we propose an inverse rendering model that estimates 3D shape, spatially-varying reflectance, homogeneous subsurface scattering parameters, and an environment illumination jointly from only a pair of captured images of a translucent object. In order to solve the ambiguity problem of inverse rendering, we use a physically-based renderer and a neural renderer for scene reconstruction and material editing. Because two renderers are differentiable, we can compute a reconstruction loss to assist parameter estimation. To enhance the supervision of the proposed neural renderer, we also propose an augmented loss. In addition, we use a flash and no-flash image pair as the input. To supervise the training, we constructed a large-scale synthetic dataset of translucent objects, which consists of 117K scenes. Qualitative and quantitative results on both synthetic and real-world datasets demonstrated the effectiveness of the proposed model. Code and Data are available at https://github.com/ligoudaner377/homo_translucent

1. Introduction

Inverse rendering is a long-standing problem in computer vision, which decomposes captured images into multiple intrinsic factors such as geometry, illumination, and material. With the estimated factors, many challenging applications like relighting [40,47,51,60,65,72], material editing [41,52] and object manipulation [29,57,62,70] become feasible. In this work, we focus on a very complicated object for inverse rendering: translucent objects. From biological tissue to various minerals, from industrial raw materials to a glass of milk, translucent objects are everywhere in our daily lives. A critical physical phenomenon in translucent objects is SubSurface Scattering (SSS): a photon penetrates the object's surface and, after multiple bounces, is eventually absorbed or exits at a different surface point. This non-linear, multiple bounces, multiple path process makes inverse rendering extremely ill-posed. To reduce the complexity of the task, we assume that the SSS parameters of the object are constant in a continuous 3-dimensional space

(homogeneous subsurface scattering). So, given a translucent object, our task is to simultaneously estimate shape, spatially-varying reflectance, homogeneous SSS parameters, and environment illumination.

However, a well-known issue of inverse rendering is the ambiguity problem—the final appearance of an object results from a combination of illumination, geometry, and material. For instance, a brighter area in the image may be due to the specular highlight. It may be very close to the light source; the coloration may also come from the light source or the object itself. The situation becomes more complicated when SSS is also considered because it is hard to tell the object intensity comes whether from the surface, subsurface, or both.

Solutions to the ambiguity problem can be roughly divided into two groups. The first group of researchers addresses the ambiguity problem by providing more information to the model. For example, some works use multi-view [5–7, 38, 59, 63, 69] or multi-light [8, 59] setup. The solution of the second group is based on various assumptions and simplifications. For example, some researchers simplify the reflectance model by assuming a Lambertian reflectance [51], some simplify the illumination by assuming that the object is illuminated by a single point light source [15], some simplify the geometry by assuming a near-planar [35]. For the SSS, most existing works [4, 8, 9, 12, 16, 34, 37, 45, 49, 50, 56, 58] simply ignore it by assuming that light does not penetrate the object's surface, and the final appearance only depends on the pure surface reflectance. On the other hand, some researchers [11, 21, 22, 28, 30, 71] consider a pure SSS without surface reflectance. Some works [14, 17, 26, 61] use the BSSRDF model to simplify SSS of optically thick materials in the form of complex surface reflectance. The limitation of these works is obvious: most translucent objects (*e.g.* wax, plastic, crystal) in our world can easily break their assumptions, as it contains both surface reflectance and SSS. We tackle a more challenging problem by considering both surface reflectance and SSS of translucent objects with an arbitrary shape under an environment illumination. However, using a complex scene representation means introduc-
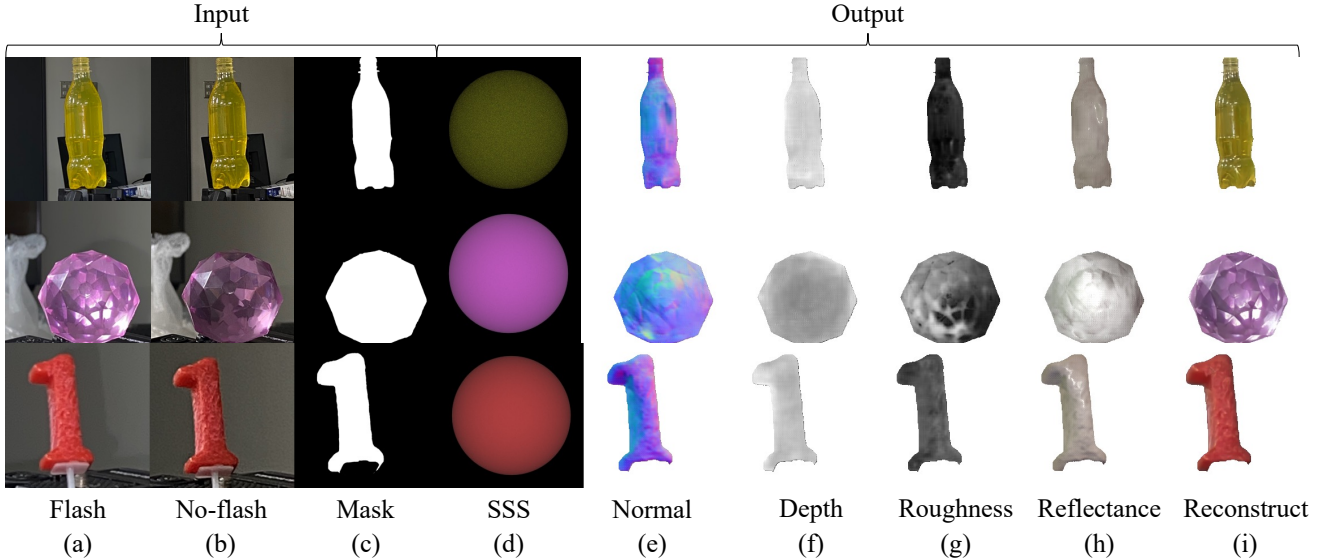
Figure 1. Inverse rendering results of real-world translucent objects. Our model takes (a) a flash image, (b) a no-flash image, and (c) a mask as input. Then, it decomposes them into (d) homogeneous subsurface scattering parameters (denoted as SSS in the figure), (e) surface normal, (f) depth, and (g) spatially-varying roughness. We use the predicted parameters to re-render the image that only considers (h) the surface reflectance, and (i) both surface reflectance and subsurface scattering. Note that (d) subsurface scattering only contains 7 parameters (3 for extinction coefficient σ_t, 3 for volumetric albedo α, 1 for phase function parameter g), we use these parameters to render a sphere using Mitsuba [46] for visualization. Brighter areas in the depth map represent a greater distance, and brighter areas in the roughness map represent a rougher surface.

ing more ambiguity.

In this work, we propose an inverse rendering framework that handles both surface reflectance and SSS. Specifically, we use a deep neural network for parameter estimation. To enable the proposed model to train and estimate the intrinsic physical parameters more efficiently, we employ two differentiable renderers: a physically-based renderer that only considers the surface reflectance of the direct illumination and a neural renderer that creates the multiple bounce illumination as well as SSS effect. Two renderers work together to re-render the input scene based on the estimated parameters and at the same time enable material editing. To enhance the supervision of the proposed neural renderer, we also propose an augmented loss by editing the SSS parameters. Moreover, inspired by the recent BRDF estimation methods [2, 8] that use a flash and no-flash set up to address the problem of unpredictability in saturated highlight, we also adopt this two-shot setup not only to deal with the saturated highlight problem but also to disentangle the surface reflectance and SSS. To train our model, we construct a synthetic dataset consisting of more than 117K translucent scenes because there is no sufficient dataset supporting translucent objects. Each scene contains a human-created 3D model and is rendered with a spatially-varying microfacet BSDF, homogeneous SSS, under an environment illumination.

Our contributions are summarized as follows:
- We first tackle the problem of estimating shape, spatially-varying surface reflectance, homogeneous SSS, and illumination simultaneously from a flash and no-flash pair of captured images at a single viewpoint.
- We build a novel model that combines a physically-based renderer and a neural renderer for explicitly separating the SSS and the other parameters.
- We introduce the augmented loss to train the neural renderer supervised by altered images whose SSS parameters were edited.
- We construct a large-scale photorealistic synthetic dataset that consists of more than 117K scenes.

2. Related Work

2.1. Inverse rendering of surface reflectance

There has been a lot of work on estimating depth [18, 31, 48], BRDF [1, 2, 15, 33, 35, 42], and illumination [19, 20, 25] separately.

With the recent rise of deep learning, more and more research is focusing on simultaneous parameter estimation. Li *et al.* [37] propose the first single-view method to estimate the shape, SVBRDF, and illumination of a single object. They use a cascaded network to tackle the ambiguity problem and an in-network rendering layer to create global

illumination. A few research groups demonstrate that a similar idea can be applied in more complex situations, such as indoor scene inverse rendering by using a Residual Appearance Renderer [50] or spatially-varying lighting [34, 54, 74]. Li *et al*. [36] extend the inverse rendering of indoor scenes by considering more complex materials like metal or mirror. Boss *et al*. [8] tackle the ambiguity problem under a saturated highlight by using a two-shot setup. Sang *et al*. [49] implement shape, SVBRDF estimation, and relighting by using a physically-based renderer and a neural renderer. Deschaintre *et al*. [16] introduce polarization into shape and SVBRDF estimation. Wu *et al*. [58] train the model in an unsupervised way with the help of the rotational symmetry of specific objects. Lichy *et al*. [39] enables a high resolution shape and material reconstruction by a recursive neural architecture. Our model can be interpreted as an extended version of these methods that also take SSS into account.

2.2. Inverse rendering of subsurface scattering

Subsurface scattering is essential for rendering translucent objects such as human skin, minerals, smoke, etc. The reconstruction and editing of these objects heavily rely on scattering parameter estimation. However, inverse scattering is a challenging task due to the multiple bounces and multiple paths of light inside the object. Some works [21, 22, 28, 30, 71] combine analysis by synthesis and Monte Carlo volume rendering techniques to tackle this problem. However, they suffer from some problems like local minimal and long optimization time.

Recently, Che *et al*. [11] used a deep neural network to predict the homogeneous scattering parameters as a warm start for analysis by synthesis. However, they do not estimate parameters such as geometry, illumination, etc. This means that their model can not handle applications like scene reconstruction, material editing. In addition, they only consider pure SSS. However, most translucent objects in our real-world consist of both surface reflectance and SSS. Our model estimates geometry, illumination, surface reflectance, and SSS simultaneously.

2.3. Differentiable rendering

Differentiable renderers are broadly used in inverse rendering field for reconstructing human faces [51, 55], indoor scenes [34, 50], buildings [43, 65], and single objects [8, 37, 49]. However, most of them use the physically-based renderer that only considers direct illumination. Such a renderer cannot produce high-quality images as they do not allow for effects like soft shadows, interreflection, and SSS. Recently, there have been some general-purpose differentiable renderers [32, 46, 66, 67] that consider indirect illumination. However, such a Monte Carlo path tracing graph takes a lot of memory and computing resources, especially when a large sample per pixel (spp) is used. Instead, some

works [37, 44, 49, 50, 64, 68] splice a neural network behind direct illumination only renderer to enable global illumination or scene editing. Inspired by them, we also designed a renderer-neural network architecture for the SSS task.

2.4. Scene editing

Scene editing is a broad research topic, and recent research advances have been dominated by deep learning. A bunch of works has been reported for relighting [40, 47, 51, 60, 65, 72], material editing [41, 52] and object manipulation [29, 57, 62, 70]. We take the first step on the scattering parameter editing of translucent objects by observing only two images.

3. Methods

In this section, we introduce the proposed inverse rendering framework. Section 3.1 presents how we model our input and output. In Section 3.2, we talk about a neural network that is used for parameter estimation. In Section 3.3 we discuss the proposed two-renderer structure. Then, we illustrate an augmented loss to enhance the supervision of the proposed neural renderer in Section 3.4. Finally, in Section 3.5 we show the loss function.

3.1. Problem setup

Scene representation For the geometry part, we use a depth map D to roughly represent the shape of the object and a normal map N to provide local details. For the surface part, we use a microfacet BSDF model proposed by [53]. A roughness map R is used to represent BSDF parameters. The homogeneous SSS is modeled by three terms: an extinction coefficient σ_t which controls the optical density, a volumetric albedo α which determines the probability of whether photons are scattered or absorbed during a volume event, and a Henyey-Greenstein phase function [24] parameter g which defines whether the scattering is forward ($g > 0$), backward ($g < 0$) or isotropic ($g = 0$). We estimate the spherical harmonics sh as a side-prediction to assist the other part of the model. A flashlight intensity i is also predicted, taking into account that the intensity of the flashlight varies from device to device.

Model design Inspired by Aittala *et al*. [2] and Boss *et al*. [8], we also use a flash and no-flash image setup. Our motivation for this design is that the visibility of translucent objects will vary at different light intensities. For example, if a bright light is placed on the back of a finger, we can clearly see the color of the blood. Such a property facilitates scattering parameter estimation and enables better disentanglement of surface reflectance and SSS.

So, given a translucent object of unknown shape, material, and under unknown illumination, our target is to estimate these parameters simultaneously. In addition, we also enable material editing by manipulating the estimated
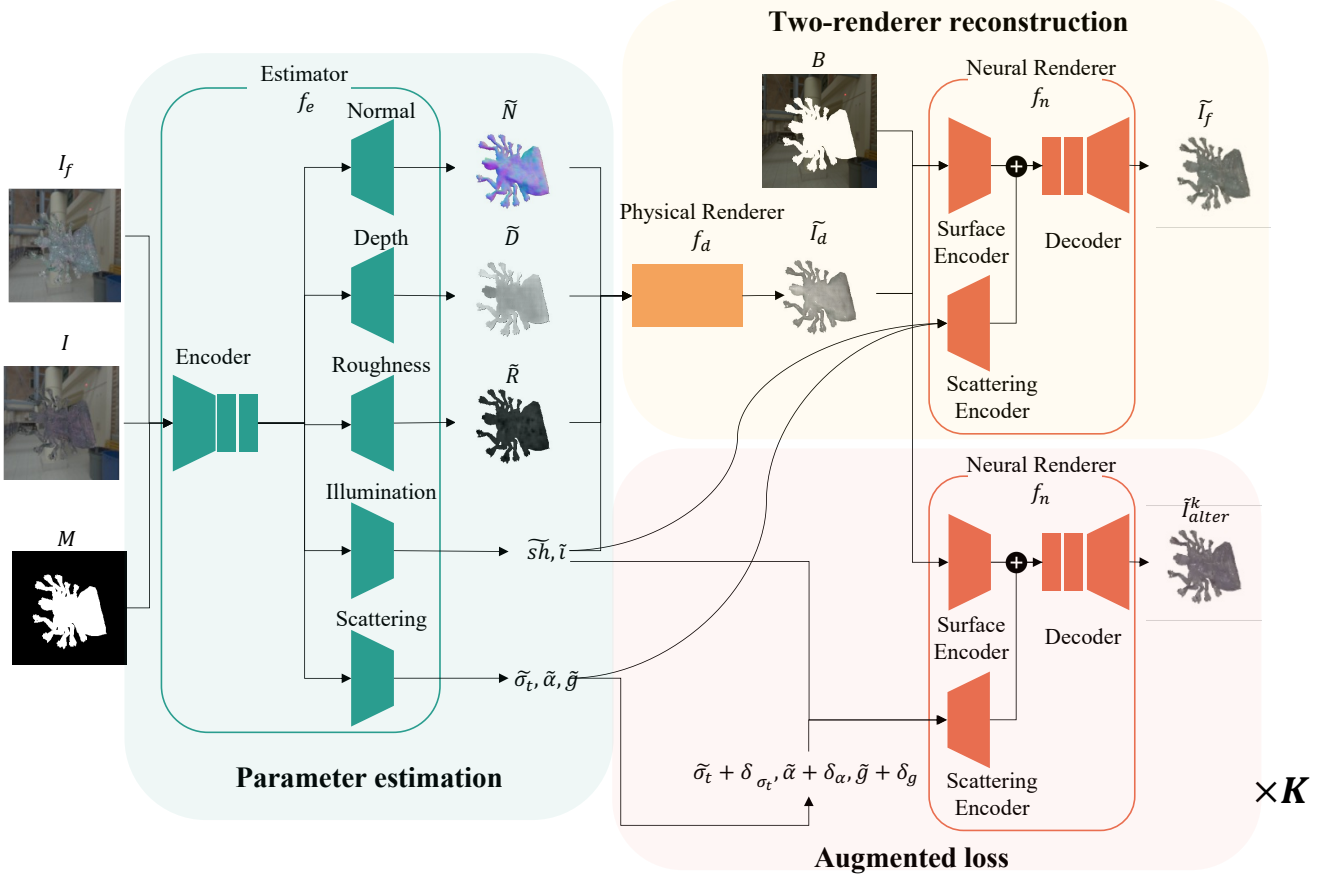
Figure 2. Overview of the proposed model. We use different colors to indicate different functions: green for the estimator, orange for the physically-based renderer, red for the neural renderer. We have K augmented loss modules and only show one in the figure.

parameters. Figure 2 shows an overview of our model. The inputs of our model are three images: a flash image $I_f \in \mathbb{R}^{3 \times 256 \times 256}$, a no-flash image $I \in \mathbb{R}^{3 \times 256 \times 256}$, and a binary mask $M \in \mathbb{R}^{256 \times 256}$. For each scene, the estimated parameters are:

- A depth map $D \in \mathbb{R}^{256 \times 256}$ and a normal map $N \in \mathbb{R}^{3 \times 256 \times 256}$ to represent the shape.
- A roughness map $R \in \mathbb{R}^{256 \times 256}$ used in the microfacet BSDF model.
- The spherical harmonics $sh \in \mathbb{R}^{3 \times 9}$ and a flashlight intensity $i \in \mathbb{R}^1$ to represent illumination.
- The extinction coefficient $\sigma_t \in \mathbb{R}^3$, volumetric albedo $\alpha \in \mathbb{R}^3$, and Henyey-Greenstein phase function parameter $g \in \mathbb{R}^1$.

3.2. Parameter estimation

Taking advantage of recent developments in deep learning, we use a deep convolutional neural network as our estimator. We use the one encoder and multiple heads structure. Our motivation for this design is that estimating shape, material, and illumination can be thought of as multi-task

learning. Ideally, each task can assist the other to learn a robust encoder. The encoder extract features from the input images and each head estimate parameters accordingly. So, given a flash image I_f, no-flash image I, and a binary mask M, the estimated physical parameters are:

$$\tilde{D}, \tilde{N}, \tilde{R}, \tilde{sh}, \tilde{i}, \tilde{\sigma}_t, \tilde{\alpha}, \tilde{g} = f_e(I, I_f, M), \qquad (1)$$

where f_e denotes the estimator. $\tilde{D}, \tilde{N}, \tilde{R}, \tilde{sh}, \tilde{i}, \tilde{\sigma}_t, \tilde{\alpha}, \tilde{g}$ are the estimated depth, normal, roughness, spherical harmonics coefficient, flashlight intensity, extinction coefficient, volumetric albedo, and Henyey-Greenstein phase function parameter.

3.3. Physical renderer and neural renderer

The use of reconstruction loss to supervise network training is a widespread technique in deep learning. Specifically, reconstruction in inverse rendering means re-rendering the scene with the estimated parameters. In addition, the rendering module must be differentiable to pass the gradient of reconstruction loss to the estimator.

One potential option is to use a general-purpose differentiable renderer such as Mitsuba [46]. However, we have two problems. First, current general-purpose differentiable renderers have memory and training speed problems for consumer-grade GPUs, especially when using a high sample rate. Because the computational cost of path tracing is much larger than that of standard neural works. For example, differentially rendering a single translucent object image with 256×256 resolution and 64 samples per pixel (spp) requires more than 5 seconds and 20 GB memory on an RTX3090 GPU. Moreover, the rendering time and memory will increase linearly when spp increase. Second, re-rendering the object with SSS requires the entire 3D shape (*e.g.*, a 3D mesh). However, it is not easy to estimate the complete 3D information at a single viewpoint. That is the reason we only estimate a normal and a depth map as our geometry representation. This representation is sufficient for rendering surface reflectance but not for SSS. Another option [8, 58] is to use a differentiable renderer that only considers direct illumination. This sacrifices some reconstruction quality but dramatically improves efficiency. However, such a method cannot be applied to a translucent object because SSS depends on multiple bounces of light inside the object.

With the assist of recent advances in image-to-image translation [27,73], many works have demonstrated the successful use of neural networks for adding indirect illumination [37, 44], photorealistic effect [43, 50], or relighting [49, 64]. Inspired by them, we propose a two-step rendering pipeline to mimic the rendering performance of SSS of a general-purpose differentiable renderer. The first step is a physically-based rendering module that only considers direct illumination:

$$\tilde{I}_d = f_d(\tilde{D}, \tilde{N}, \tilde{R}, \tilde{i}, \tilde{sh}), \qquad (2)$$

where f_d is a physically-based renderer that follow the implementation of [49], and it has no trainable parameters. $\tilde{I}_d$ is the re-rendered image that only consider the surface reflectance. The second step is a neural renderer f_n to create SSS effect:

$$\tilde{I}_f = f_n(\tilde{I}_d, \tilde{sh}, \tilde{i}, \tilde{\sigma}_t, \tilde{\alpha}, \tilde{g}, B), \qquad (3)$$

where $\tilde{I}_f$ is the re-rendered flash image. f_n is the proposed neural renderer, and it consists of 3 parts (See Figure 2 for reference), a Surface encoder, a Scattering encoder, and a decoder. The Surface encoder maps the estimated surface reflectance image $\tilde{I}_d$ into a feature map. In addition to the surface reflectance image, we also input the background image B (masked-out version of I_f) to the Surface encoder to provide high frequency illumination information. The Scattering encoder consists of a few upsampling layers. It maps $\tilde{sh}, \tilde{i}, \tilde{\sigma}_t, \tilde{\alpha}, \tilde{g}$ to a feature map. The decoder consists of some Resnet blocks [23] and upsampling layers. The task of our neural renderer can be considered as a conditional image-to-image translation, where the condition is the SSS parameters.

The advantage of the two-renderer design is that it is naturally differentiable. At the same time, the training cost is acceptable. In addition, the physically-based renderer can provide physical hints to the neural renderer. Because we separate the reconstruction of surface reflectance and SSS explicitly, the problem of ambiguity can be alleviated.

3.4. Augmented loss

In this subsection, we present an augmented loss to address the hidden information problem by using multiple altered images to supervise the proposed neural renderer. In Section 3.3, we propose a two-renderer structure to compute the reconstruction images to improve the parameter estimation. However, a well-known problem of reconstruction loss in deep learning is that neural networks learn to "hide" information within them [13]. This may cause our neural renderer to ignore the estimated SSS parameters and only reconstruct the input image by the hidden information. If so, the reconstruction loss cannot give correct gradients and thus fail to guide the training of the estimator.

In order to solve this problem, we enhance the supervision of the proposed neural renderer by an augmented loss. Specifically, after the parameters are estimated, we edit the estimated SSS parameters and let the Neural renderer reconstruct image based on the edited SSS parameters. We also render K altered images I_{alter}^k as their ground truth labels to train the neural renderer. The altered images have the same parameters as the original flash image, except the SSS parameters. Specifically, I_f and I_{alter}^k share the same shape, surface reflectance, and illumination, but different extinction coefficient, phase function, and volumetric albedo. The edited SSS parameters are randomly sampled from the same distribution as the original ones:

$$\tilde{I}_{alter}^k = f_n(\tilde{I}_d, \tilde{sh}, \tilde{i}, \tilde{\sigma}_t + \delta_{\sigma_t}, \tilde{\alpha} + \delta_\alpha, \tilde{g} + \delta_g, B), \quad (4)$$

where $\delta_{\sigma_t}, \delta_\alpha, \delta_g$ are the differences between target SSS parameters (subsurface scattering parameters of I_{alter}^k) and the estimated ones, $\tilde{I}_{alter}^k$ is the reconstructed altered images. The benefits of this design are as follows. First, the input image is not the same as the image to be reconstructed, which makes it meaningless for the neural network to "hide" the information of the original input image. Second, the variety of input parameters and output images makes the neural renderer more sensitive to the changes in SSS parameters. These two points make the neural renderer a good guide for the estimator, which allows for a more accurate estimation of the SSS parameters. Considering the time and computing resources it takes to render the training images, in practice, we set $K = 3$.

| Flash | No-flash | Mask | Normal | Depth | Roughness | Reflectance | SSS | Spherical harmoincs | Altered 1 | Altered 2 | Altered 3 |

Figure 3. Examples of the proposed dataset, we use the ground truth shape and illumination to visualize SSS parameters.

3.5. Loss functions

The proposed model is fully supervised and trained end to end, and we compute the loss between the estimated parameters their ground truths:

$$
\begin{aligned}
L &= L_D + L_N + L_R + L_{sh} + L_i \qquad (5) \\
&+ L_{\sigma_t} + L_\alpha + L_g + L_{I_f} + \sum_K L_{alter}^k,
\end{aligned}
$$

where $L_D, L_N, L_R, L_{I_f}, L_{alter}^k$ stand for the $L1$ loss between the estimated depth, normal, roughness, flash image, altered images and their ground truth ones. $L_{sh}, L_i, L_{\sigma_t}, L_\alpha, L_g$ stands for the $L2$ loss between the estimated spherical harmonics, flashlight intensity, extinction coefficient, volumetric albedo, Henyey-Greenstein phase function parameter and their ground truth ones.

4. Experiments

We introduce our dataset in Section 4.1. In Section 4.2, we compare our model with Che *et al*. [11]. We conduct an ablation study and report the quantitative and qualitative results in Section 4.3. In Section 4.4 we show the results of SSS parameter editing application. More experiment results can be found in the supplementary document.

4.1. Datasets

In this work, we propose the first large-scale synthetic translucent dataset that contains both surface reflectance and SSS. We collected the 3D objects from ShapeNet [10], human-created roughness map and auxiliary normal map from several public resources, and environment map from the Laval Indoor HDR dataset [19].

For each scene, we used Mitsuba [46] to render five images: flash image, no-flash image, and three altered images. Each scene contains a randomly selected object, auxiliary normal map, roughness map, environment map. We also randomly sample the SSS parameters and flash light intensity. Similar to some SfP (Shape from Polarization) works [3], we assume a constant IoR (Index of Refraction) to reduce the ambiguity problem. A full introduction to our dataset can be found in the supplementary documentation.

4.2. State-of-the-art comparison

To the best of our knowledge, we are the first to tackle the inverse rendering problem of translucent objects containing both surface reflectance and SSS. It is not easy to compare our method with **pure surface reflectance** methods [8, 37]. Although similar to our method, these methods also predict parameters like surface normal, depth, and roughness. The key problem is that the coloration of pure surface reflectance models is affected by "diffuse albedo", which is a parameter of BRDF. However, in our proposed scene representation, the surface reflection and refraction are modeled by BSDF, and the SSS is modeled by the Radiative Transport Equation (RTE), which means that the coloration is affected by the volumetric albedo, extinction coefficient, and phase function. In conclusion, training pure surface reflectance methods on our dataset is impossible. Thus, we choose a **pure SSS** method proposed by Che *et al*. [11] to compare with our model. Their method requires an edge map as the additional input of the neural network. We follow their paper to generate the edge maps for our dataset. We train their model on our dataset with the same hyperparameters as our method. We report the Mean Absolute Error (MAE) in Table 1, and visual comparison in Figure 4. It is difficult to compare the parameter estimation accuracy without GT parameter references, so we only show the synthetic data results. It is observed that their method fails to estimate reasonable SSS parameters due to the highly ambiguous scene representation.

4.3. Ablation study

We conduct ablation studies to evaluate the effects of each component of the proposed model. Specifically, we start with a Baseline model, which uses the same estimator as our Full model, but only reconstructs the input scene with a neural network. So, the baseline model is essentially an autoencoder. Then, we divide the reconstruction into two steps: a physically-based renderer that reconstructs the surface reflectance and a neural renderer that create the multibounce illumination as well as the SSS effect. We call this model "2R". After that, we introduce the augmented loss to the "2R" model and denote it as "2R-AUG". Finally, we implement the Full model by adding the two-shot setting

Table 1. MAE results on 17140 test scenes. For each element we report mean(std) value. The scale of mean is 1×10^0, and std is 1×10^{-3}.

|	Geometry		BSDF	Illumination		SSS		
	N	D	R	sh	i	σ_t	α	g
Baseline	.0918(.4395)	.0705(.4443)	.0811(.5303)	.1083(.6571)	.0912(1.042)	.1670(.7904)	.1061(.5792)	.1762(.8811)
2R	.0916(.3009)	.0697(.3617)	.0811(.4903)	.1064(.8984)	.0908(.7697)	.1675(1.072)	.1057(.4191)	.1777(2.562)
2R-AUG	.0913(.1768)	.0699(1.271)	.0807(.2714)	.1105(8.926)	.0893(1.151)	.1619(.9635)	.1040(.1578)	.1703(.8856)
Che *et al.* [11]	-	-	-	-	-	.1828	.1115	.2123
Full model	**.0894**(.1532)	**.0646**(.3283)	**.0769**(.2659)	**.0989**(1.017)	**.0804**(1.736)	**.1590**(.2286)	**.1002**(.5185)	**.1655**(.3932)
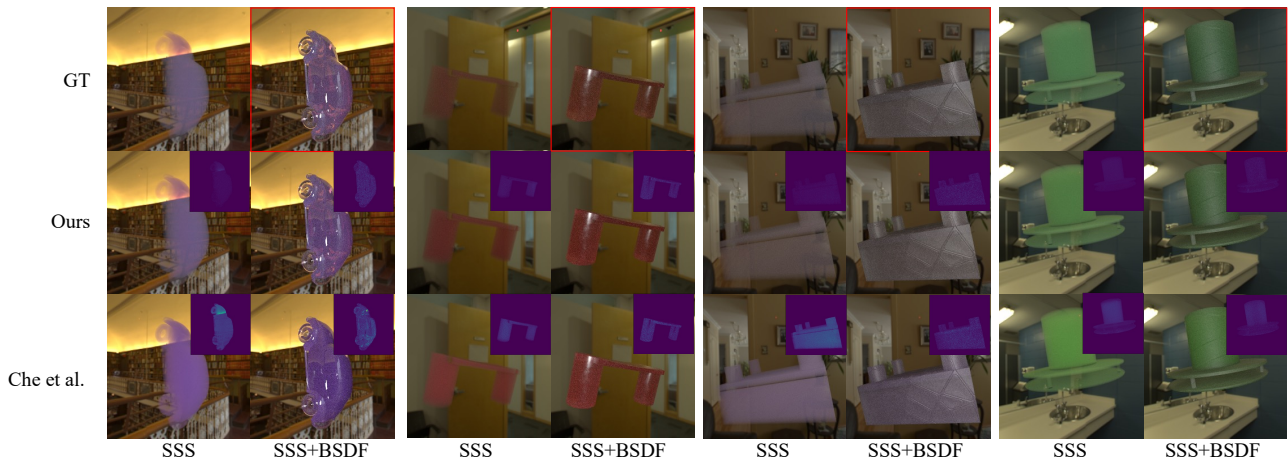
Figure 4. Visual comparison with Che *et al.* [11]. We use two visualizations which are "SSS" and "SSS+BSDF" to show the parameter estimation results. "SSS" is rendered by the GT illumination, shape, and predicted SSS. "SSS+BSDF" is rendered by the GT illumination, shape, BSDF, and predicted SSS. Images in red rectangles are input images. Error maps are in the upper right corner.

to "2R-AUG". We report the MAE results on the synthetic data of all experiments in Table 1. It is observed that for most of the metrics, "2R" model outperforms the Baseline model. Especially for the illumination part, the accuracy improved a lot. This confirms our previous point that is explicitly separating the surface reflectance, and SSS reduces ambiguity. Although the performance of "2R" and "2R-AUG" are similar for geometry, illumination, and surface reflectance parts, the result of SSS parameters which are enhanced by the augmented loss, is improved. Comparison between the "2R-AUG" and Full model proves that the proposed two two-shot problem settings can further reduce the ambiguity problem. Figure 5 demonstrate some visual comparisons. It is observed that the SSS images of the Full model are most similar to the ground truth ones, and the other models are unstable. In addition, because of the two-shot setting, the full model's estimation of environment illumination is also more accurate. To show the flexibility of the proposed model, we also test on several common real-world translucent objects, and illustrate the results in Figure 1. The results show that our model can estimate reasonable SSS parameters.

4.4. Material editing

Given a translucent object with an unknown shape, illumination, and material, we show that the proposed inverse rendering framework can edit translucent objects based on the given SSS parameters. Figure 6 shows the editing results.

5. Conclusions and Limitations

In this paper, we made the first attempt to jointly estimate shape, spatially-varying surface reflectance, homogeneous SSS, and illumination from a flash and no-flash image pair. The proposed two-shot, two-renderer, and augmented loss reduced the ambiguity of inverse rendering and improved the parameter estimation. In addition, we also constructed a large-scale synthetic dataset of fully labeled translucent objects. The experiments of the real-world dataset demonstrated that our model could be applied to images captured by a smartphone camera. Finally, we demonstrated that our pipeline is also capable of SSS parameter editing.

The proposed method still has some limitations. First, to reduce the ambiguity problem, we set the IoR to be constant and do not estimate it. However, IoR affects light transmission through object boundaries and thus in-
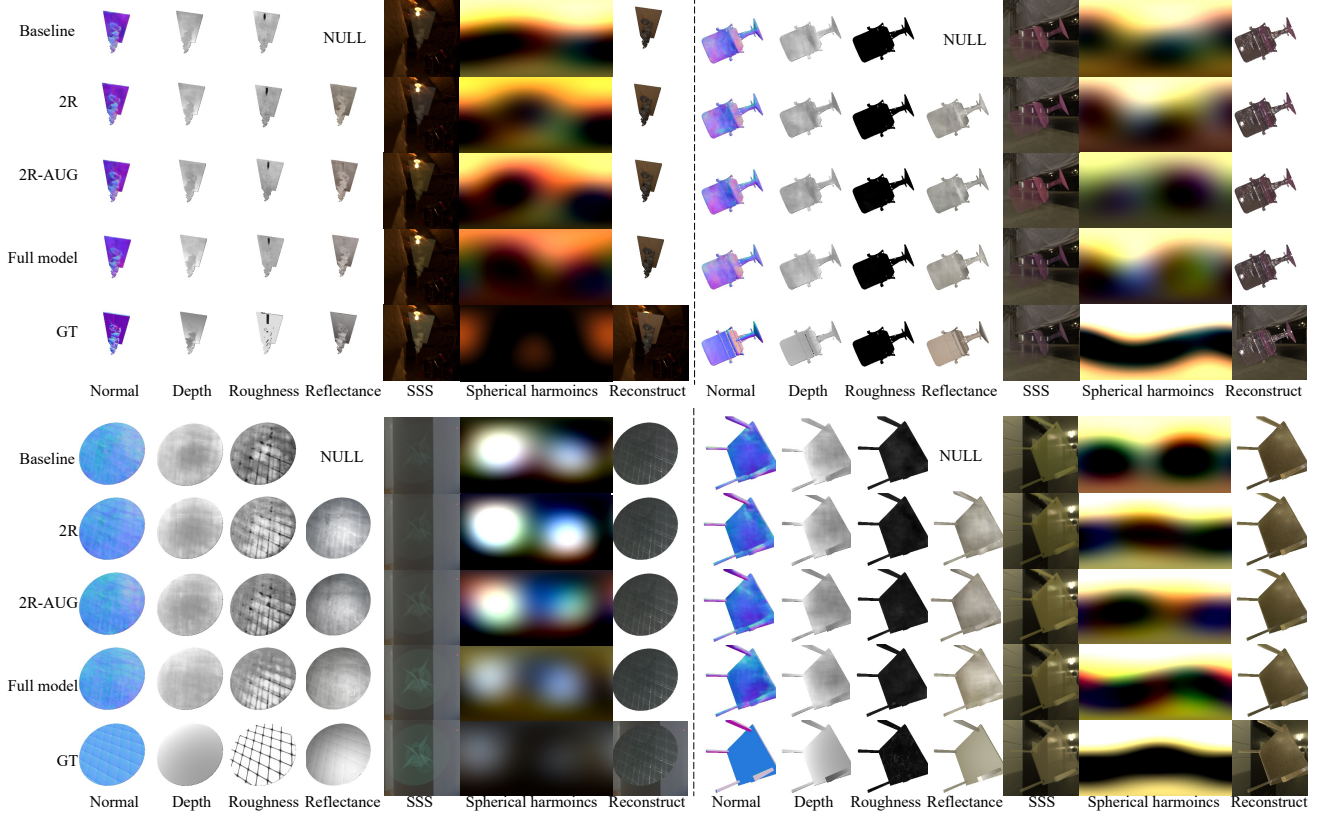
Figure 5. Visual results of ablation study on the synthetic dataset. For visualization, we use the ground truth shape and illumination to render the image of the estimated SubSurface Scattering parameters (denoted as SSS in the figure).

Figure 6. SSS parameter editing results. The 3rd row is the input images. We randomly edit the scattering parameters before inputting them into the neural renderer. The 2nd row is the estimated images, and we show their ground truth images in the 1st row.

fluences the SSS. For some materials with large or small IoR, our model may not work. Second, our model does not support relighting and novel-view synthesis. Unlike pure surface reconstruction models that the estimated normals can be easily applied to physically-based renderers for relighting or novel-view synthesis, rendering translucent objects requires a full 3D estimation (*e.g.*, mesh) including the backside information. However, estimating the complete geometry is difficult for single-view reconstruction method. Solving these challenges can be good future work.

References

[1] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 2

[2] Miika Aittala, Tim Weyrich, Jaakko Lehtinen, et al. Two-shot svbrdf capture for stationary materials. *ACM Trans. Graph.*, 34(4):110–1, 2015. 2, 3

[3] Yunhao Ba, Alex Gilbert, Franklin Wang, Jinfa Yang, Rui Chen, Yiqin Wang, Lei Yan, Boxin Shi, and Achuta Kadambi. Deep shape from polarization. In *European Conference on Computer Vision*, pages 554–571. Springer, 2020. 6

[4] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2014. 1

[5] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5960–5969, 2020. 1

[6] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[7] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[8] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik Lensch, and Jan Kautz. Two-shot spatially-varying brdf and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2020. 1, 2, 3, 5, 6

[9] Mark Boss and Hendrik P.A. Lensch. Single image brdf parameter estimation with a conditional adversarial network. In *ArXiv e-prints*, 2019. 1

[10] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 6

[11] Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Towards learning-based inverse subsurface scattering. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020. 1, 3, 6, 7

[12] Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khamis, Or Litany, and Sanja Fidler. Dib-r++: Learning to predict lighting and material with a hybrid differentiable renderer. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[13] Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017. 5

[14] Xi Deng, Fujun Luan, Bruce Walter, Kavita Bala, and Steve Marschner. Reconstructing translucent objects using differentiable rendering. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 1

[15] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (ToG)*, 37(4):1–15, 2018. 1, 2

[16] Valentin Deschaintre, Yiming Lin, and Abhijeet Ghosh. Deep polarization imaging for 3d shape and svbrdf acquisition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15567–15576, 2021. 1, 3

[17] Bo Dong, Kathleen D Moore, Weiyi Zhang, and Pieter Peers. Scattering parameters and surface normals from homogeneous translucent materials using photometric stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2291–2298, 2014. 1

[18] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 2

[19] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090*, 2017. 2, 6

[20] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. What is around the camera? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5170–5178, 2017. 2

[21] Ioannis Gkioulekas, Anat Levin, and Todd Zickler. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision*, pages 685–701. Springer, 2016. 1, 3

[22] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG)*, 32(6):1–13, 2013. 1, 3

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[24] L. G. Henyey and J. L. Greenstein. Diffuse radiation in the Galaxy. *The Astrophysical Journal*, 93:70–83, Jan. 1941. 3

[25] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7312–7321, 2017. 2

[26] Chika Inoshita, Yasuhiro Mukaigawa, Yasuyuki Matsushita, and Yasushi Yagi. Surface normal deconvolution: Photometric stereo for optically thick translucent objects. In *European*

Conference on Computer Vision, pages 346–359. Springer, 2014. 1

[27] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 5

[28] Pramook Khungurn, Daniel Schroeder, Shuang Zhao, Kavita Bala, and Steve Marschner. Matching real fabrics with micro-appearance models. *ACM Trans. Graph.*, 35(1):1–1, 2015. 1, 3

[29] Tejas D Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B Tenenbaum. Deep convolutional inverse graphics network. *arXiv preprint arXiv:1503.03167*, 2015. 1, 3

[30] Aviad Levis, Yoav Y Schechner, Amit Aides, and Anthony B Davis. Airborne three-dimensional cloud tomography. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3379–3387, 2015. 1, 3

[31] Ruibo Li, Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, and Lingxiao Hang. Deep attention-based classification network for robust depth prediction. In *Asian Conference on Computer Vision*, pages 663–678. Springer, 2018. 2

[32] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 3

[33] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics (ToG)*, 36(4):1–11, 2017. 2

[34] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan 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*, pages 2475–2484, 2020. 1, 3

[35] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: Svbrdf acquisition with a single mobile phone image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 72–87, 2018. 1, 2

[36] Zhen Li, Lingli Wang, Xiang Huang, Cihui Pan, and Jiaqi Yang. Phyir: Physics-based inverse rendering for panoramic indoor images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12713–12723, 2022. 3

[37] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 1, 2, 3, 5, 6

[38] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: neural 3d reconstruction of transparent shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1262–1271, 2020. 1

[39] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. Shape and material capture at home. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6123–6133, 2021. 3

[40] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A Efros, and Noah Snavely. Learning to factorize and relight a city. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 544–561. Springer, 2020. 1, 3

[41] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2261–2269, 2017. 1, 3

[42] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6315–6324, 2018. 2

[43] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 3, 5

[44] Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, H-P Seidel, and Tobias Ritschel. Deep shading: convolutional neural networks for screen space shading. In *Computer graphics forum*, volume 36, pages 65–78. Wiley Online Library, 2017. 3, 5

[45] Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 1

[46] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)*, 38(6):1–17, 2019. 2, 3, 5, 6

[47] Rohit Pandey, Sergio Orts Escolano, Chloe Legendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello. Total relighting: learning to relight portraits for background replacement. *ACM Transactions on Graphics (TOG)*, 40(4):1–21, 2021. 1, 3

[48] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 2

[49] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *European Conference on Computer Vision*, pages 85–101. Springer, 2020. 1, 3, 5

[50] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 1, 3, 5

[51] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6296–6305, 2018. 1, 3

[52] Jian Shi, Yue Dong, Hao Su, and Stella X Yu. Learning non-lambertian object intrinsics across shapenet categories.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1694, 2017. 1, 3

[53] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. *Rendering techniques*, 2007:18th, 2007. 3

[54] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3d spatially-varying lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12538–12547, October 2021. 3

[55] Yandong Wen, Weiyang Liu, Bhiksha Raj, and Rita Singh. Self-supervised 3d face reconstruction via conditional estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13289–13298, 2021. 3

[56] Felix Wimbauer, Shangzhe Wu, and Christian Rupprecht. De-rendering 3d objects in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18490–18499, 2022. 1

[57] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural scene de-rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 699–707, 2017. 1, 3

[58] Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. De-rendering the world's revolutionary artefacts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6338–6347, 2021. 1, 3, 5

[59] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 38(4):1–13, 2019. 1

[60] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)*, 37(4):1–13, 2018. 1, 3

[61] Jingjie Yang and Shuangjiu Xiao. An inverse rendering approach for heterogeneous translucent materials. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*, pages 79–88, 2016. 1

[62] Shunyu Yao, Tzu Ming Harry Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, William T Freeman, and Joshua B Tenenbaum. 3d-aware scene manipulation via inverse graphics. *arXiv preprint arXiv:1808.09351*, 2018. 1, 3

[63] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *arXiv preprint arXiv:2003.09852*, 2020. 1

[64] Ye Yu, Abhimitra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and William AP Smith. Self-supervised outdoor scene relighting. In *European Conference on Computer Vision*, pages 84–101. Springer, 2020. 3, 5

[65] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019. 1, 3

[66] Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. Path-space differentiable rendering. *ACM Trans. Graph.*, 39(4):143:1–143:19, 2020. 3

[67] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38(6):227:1–227:16, 2019. 3

[68] Cheng Zhang, Zihan Yu, and Shuang Zhao. Path-space differentiable rendering of participating media. *ACM Transactions on Graphics (TOG)*, 40(4):1–15, 2021. 3

[69] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 1

[70] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv preprint arXiv:2010.09125*, 2020. 1, 3

[71] Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. Downsampling scattering parameters for rendering anisotropic media. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016. 1, 3

[72] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7194–7202, 2019. 1, 3

[73] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 5

[74] Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker. Irisformer: Dense vision transformers for single-image inverse rendering in indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2822–2831, 2022. 3