Two-shot Spatially-varying BRDF and Shape Estimation

Mark Boss[1*], Varun Jampani[2], Kihwan Kim[2], Hendrik P.A. Lensch[1], Jan Kautz[2]

[1]University of Tübingen, [2]NVIDIA

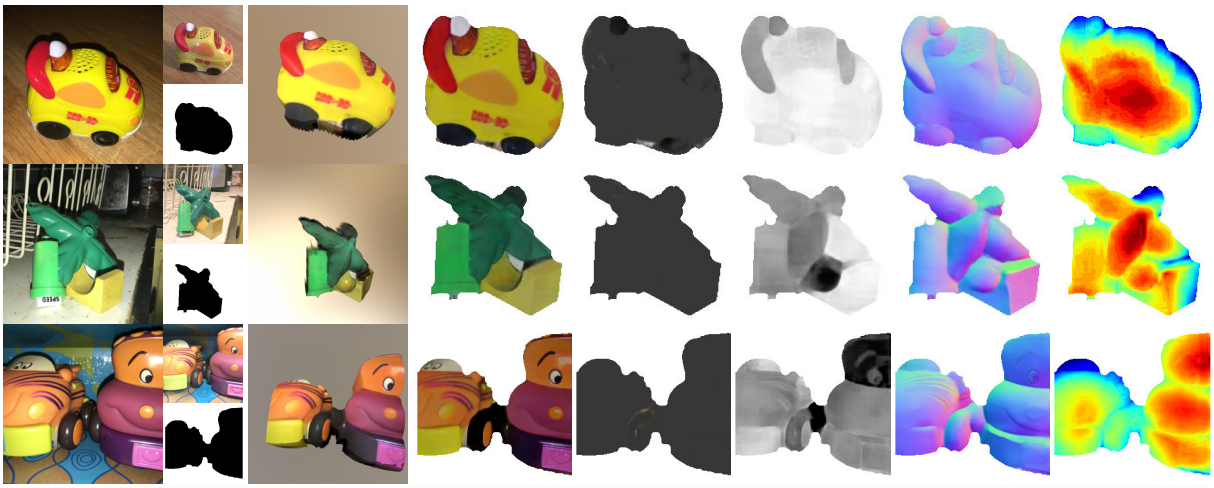

Figure 1: **Practical SVBRDF and shape estimation.** Sample two-shot input and the corresponding estimates for SVBRDF (albedo, specularity, roughness) and shape (depth and normals). Samples are taken from [5].

Abstract

Capturing the shape and spatially-varying appearance (SVBRDF) of an object from images is a challenging task that has applications in both computer vision and graphics. Traditional optimization-based approaches often need a large number of images taken from multiple views in a controlled environment. Newer deep learning-based approaches require only a few input images, but the reconstruction quality is not on par with optimization techniques. We propose a novel deep learning architecture with a stage-wise estimation of shape and SVBRDF. The earlier predictions guide each estimation, and a joint refinement network later refines both SVBRDF and shape. We follow a practical mobile image capture setting and use unaligned two-shot flash and no-flash images as input. Both our two-shot image capture and network inference can run on mobile hardware. We also create a large-scale synthetic training dataset with domain-randomized geometry and realistic materials. Extensive experiments on both synthetic and real-world datasets show that our networks trained on a synthetic dataset can generalize well to real-world images. Comparisons with recent approaches demonstrate the superior performance of the proposed approach.

1. Introduction

The estimation of intrinsic attributes of a scene such as shape and reflectance of objects and the illumination condition of the scene is often called as an *inverse rendering problem* in computer vision [51, 45, 23], and has been a core of many applications such as relighting of images [46], photo-realistic mixed reality [39], and automatic creation of assets for content creation tasks [4].

In this work, we are interested in the automatic estimation of the shape and appearance of the object in a scene from only two images. In particular, we represent the shape of the object with a depth map and the appearance as a Bidirectional Reflectance Distribution Function (BRDF) [43]. A BRDF describes the low-level material properties of an object that defines how light is reflected at any given point on an object surface. One of the most popular parametric

*Work done during an internship at NVIDIA.

Dataset and Code available at: markboss.me/publication/cvpr20-two-shot-brdf

models [12] represents the diffuse and specular properties and the roughness of the surfaces. Since the material properties can vary across the surface, one has to estimate the BRDF at each image pixel for a more realistic appearance (i.e., spatially-varying BRDF (SVBRDF)).

As the BRDF is dependent on view and light directions and estimating depth from a single 2D image is an ambiguous task, multi-view setups improve the estimation accuracy of both shape [50] and BRDF [39]. Predicting shape and BRDF from only a few images is still very challenging. For shape estimation, the advances in deep learning-based depth estimation allow us to estimate the depth of a single [17, 25], or a pair of images [58] efficiently. As monocular depth estimation is not as accurate as multi-view approaches, we exploit shading cues on the surface to disambiguate the geometric shape [6, 65] in our approach.

We propose a neural network-based approach to estimate SVBRDF and shape of an object along with the illumination from given two-shot images: flash and no-flash pairs. Some recent deep learning approaches [14, 34, 35] for BRDF estimation use only a single flash image as input. Flash images often have harsh reflective highlights where the input pixel information is saturated in non-HDR images.

Li *et al*. [35] uses a single input image and estimates shape and part of the BRDF, such as diffuse albedo and the roughness while ignoring the specular color. In this work, we use flash and no-flash image pairs as input allowing the network to access pixel information from the no-flash image when the corresponding pixels are saturated in the flash image. We focus on practical utility: Our input capture setup follows a real-world scenario where the two-shot images are consecutively taken using a mobile phone camera in burst capture. The system is designed to tackle the misalignment between the two-shot images due to camera shake.

A pivotal challenge for any learning approach is the need for training data. We tackle this issue by creating a large-scale synthetic dataset. Flash and no-flash images are rendered using high-quality, human-authored SVBRDF textures that are applied to synthetic geometry generated by domain randomization [55] of geometric shapes and backgrounds. Our networks trained on this synthetic data generalize well to real-world object images.

Another key challenge in shape and SVBRDF estimation is the problem of ambiguity. For example, a darker region in an image could be created by its material color being dark, the area slightly shadowed due to its shape, or the illumination at that spot being darker. We tackle this ambiguity by using a cascaded approach, where separate neural networks are used to estimate shape (depth), illumination, and SVBRDF. Specifically, we first estimate depth and normals using a geometry estimation network. Then the illumination is approximated, followed by SVBRDF reconstruction. Each step is guided by the estimates of the previous networks. Finally, shape and SVBRDF are optimized jointly using a refinement network. Each task is implemented by specialized network architectures. Empirically, this cascaded regression approach works reliably better compared to a single-step joint estimation. As a favorable side-effect of this cascaded approach, the size of each network is small compared to a large joint estimation network. This allows the inference networks to even operate on a mobile device. Coupled with two-shot mobile capturing, this presents a highly practical application.

Quantitative analysis based on a synthetic dataset comprising of realistic object shapes and SVBRDFs demonstrates that our approach produces more accurate estimates of shape and SVBRDF compared to baseline approaches. We also qualitatively demonstrate the applicability of our approach on a real-world two-shot dataset [5].

2. Related work

The literature on object SVBRDF and/or shape estimation is vast. Here, we only discuss the representative works that are related to ours.

BRDF Estimation An exhaustive sampling of each BRDF dimension demands long acquisition times. Several proposed methods focus on reducing acquisition time [27, 3, 16]. These methods introduce capture setups and optimization techniques that reduce the number of images required to reconstruct high-quality SVBRDF. Recently, several attempts [14, 31, 34, 2, 4] reconstruct the SVBRDF on flat surfaces with one or two flash images. These approaches leverage neural networks trained on large amounts of data and resolve the problem of ambiguity to some extent by learning the statistical properties of BRDF parameters.

For a joint estimation of shape and shading, separate optimization steps for shape and shading are common [26, 40, 19, 7]. Lensch *et al*. [26] introduce Lumitexels, which stack previously acquired shape information with the luminance information from the input images, to guide the BRDF estimation and to reduce ambiguities in the optimization. Compared to a joint estimation, fewer local minima are found, and the optimization is more robust. Recently, the task of predicting the shape and BRDF of objects or scenes is also addressed using deep learning models [35, 51]. Li *et al*. [35] predict the shape and BRDF of objects from a single flash image using an initial estimation network followed by several cascaded refinement networks. Here, the BRDF consists of diffuse albedo and specular roughness but lacks the specular albedo. Specularity is, however, essential in re-rendering metallic objects, for example.

Compared to Li *et al*. [35], our method additionally estimates the SVBRDF with specular albedo. In comparison to flat surface SVBRDF estimation [14, 34, 2, 4], our method handles full objects with shape from any view position. Additionally, due to our unaligned two-shot setup, saturated

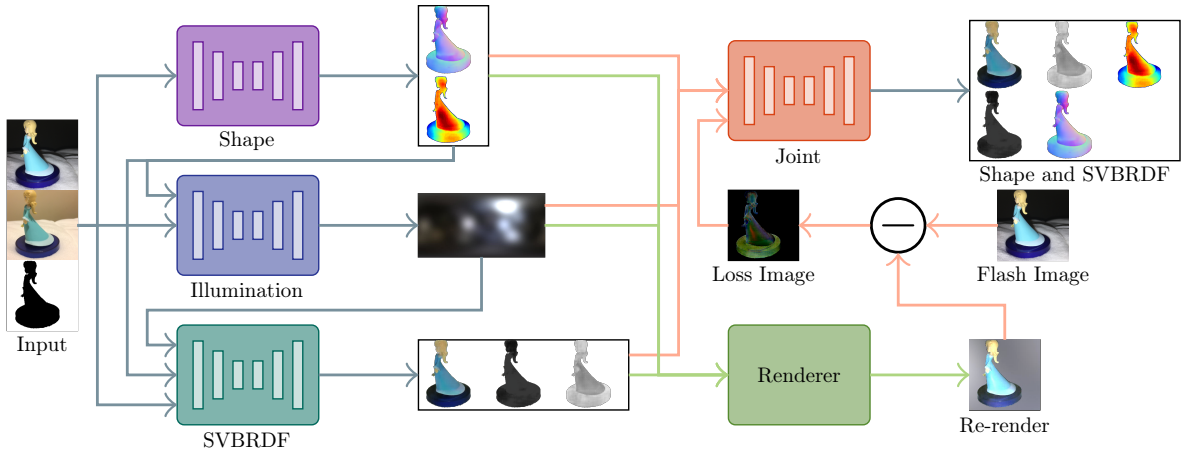

Figure 2: **Cascaded Network.** Overview of the inference pipeline for shape, illumination and SVBRDF estimation.

flash highlights are better compensated, while still providing the same one-button press capture experience for the user, due to our mobile capture scenario.

Intrinsic Imaging Intrinsic imaging is the task of decomposing an image of a scene into reflectance (diffuse albedo), and shading [8, 7, 38, 53]. With the advance in deep learning, the problem of separating shape, reflectance, and shading is tackled from labeled data [28, 41, 52], unlabeled [33] and partially labeled data [66, 32, 42, 9]. Due to the very simplistic rendering model, the use cases are limited compared to our SVBRDF estimation setup, which can be used for general re-rendering in new light scenarios.

Shape Estimation One can obtain high-quality depth from stereo images, but the problem of monocular depth estimation is quite challenging. Monocular depth estimation is predominantly tackled with deep learning [61, 37, 30, 18, 49, 25] in the recent years. This problem is especially challenging as no absolute scale is known from single images, and the depth cues need to be resolved by shading information such as the quadratic light fall-off [36].

3. Methods

As briefly discussed in the introduction, to tackle the problem of ambiguity in shape and SVBRDF estimation, we propose a novel cascaded network design for shape, illumination, and SVBRDF predictions. Fig. 2 shows an overview of our cascaded network.

Problem Setup Our network takes two-shot object images (flash and no-flash) with the corresponding foreground object mask and estimates shape and SVBRDF. We also estimate illumination as a side-prediction to help shape and SVBRDF prediction. The two-shot images can be slightly misaligned to support practical image capture with a handheld camera. The object mask allows us to evaluate only the pixels of the object in the flash image and is easily generated with GrabCut [48]. The object shape is represented as depth and normal at each pixel. The depth map provides a rough shape of the object, while the normal map models local changes more precisely. This shape representation is commonly used in various BRDF estimation methods [35, 40]. We use the Cook-Torrence model [12] to represent the BRDF at each pixel with diffuse albedo (3 parameters), specular albedo (3), and roughness (1). Similar to [60, 29], we estimate the environment illumination with 24 spherical Gaussians.

Network Overview and Motivation In order to tackle the shape/SVBRDF ambiguity, we take the inspiration from traditional optimization techniques [26, 40], which iteratively minimize a residual and alternate between optimizing for shape and/or reflectance. Thus, separate networks are used for shape, illumination, and SVBRDF estimation in a cascaded as well as an iterative manner. Predictions from earlier stages of the networks in the cascade are used as inputs to later networks to guide network predictions to better solutions. In addition, the scene is re-rendered with the current estimates, and refined further using the residual image.

Since flash and no-flash images are slightly misaligned, shape estimation is less challenging compared to SVBRDF estimation. Mis-alignment in pixels, as well as pixel differences between two-shot images [36], are a good indicator of object depth. Thus, we first predict depth and normals using a specialized merge convolutional network followed by a shape-guided illumination estimation. Then, the SVBRDF is predicted with the current estimates of shape and illumination as additional input. Finally, after computing a residual image, we refine both shape and SVBRDF using a joint refinement network. Refer to the supplementary for network architecture details.

3.1. Shape Estimation with Merge Convolutions

Since the camera parameters are unknown and the two-shot images have a minimal baseline, traditional structure-

from-motion or stereo solutions are not useful for dense depth estimation. The shape estimation needs to rely on the unstructured perspective shift as well as pixel differences between flash and no-flash images. In order to tightly integrate information from both the images, we design a specialized convolutional network for shape estimation.

For depth and normal map prediction, we use a U-net like encoder-decoder architecture [47]. Instead of standard convolution blocks, we propose to use novel merge convolution blocks (MergeConv). We concatenate the object mask with each of the two-shot input images as input to the network. Fig. 3 illustrates the MergeConv block. Both the input images or their intermediate features are separately processed by 2D convolutions (Conv2D). The outputs of each Conv2D operation are concatenated in channels with the merged output from the previous MergeConv layer and is processed with another Conv2D operation. Inspired by residual connections in ResNet [20], we add the Conv2D outputs as indicated in Fig. 3. We use 4 MergeConv blocks for the encoder and also 4 for the decoder. During encoding, max pooling for $2\times$ spatial downsampling is used. For each MergeConv in the decoder, we use $2\times$ nearest neighbor upsampling. The final depth and normal map estimates are produced using a separate 2D convolution, followed by a sigmoid activation. The rationale behind this MergeConv architecture is to keep separating the process of pathways for both the input images while exchanging (merging) the information between them using a third pathway in the middle. We believe that information in both input images is essential for shape reasoning, and this architecture helps to keep the features from each of the images intact throughout the network. Empirically, we observe reliably better shape predictions with this architecture compared to a standard U-net with a similar number of network parameters.

Training losses are based on the $\mathcal{L}_2$ distance between ground-truth (GT) and predicted depths, $\mathcal{L}_2^{\text{depth}}$, as well as the angular distance between GT and predicted normals, $\mathcal{L}_{angular}^{\text{normals}}$. Besides, we use a new consistency loss between the predicted normal $\mathbf{n}$ and a normal $\mathbf{n}^*$ derived from the depth information $\mathbf{d}$, which enforces that the predicted normals follow the curvature of the shape:

$$\mathcal{L}_{consistency}^{\text{normals/depth}} = \frac{\mathbf{n}}{\|\mathbf{n}\|} - \frac{\mathbf{n}^*}{\|\mathbf{n}^*\|}, \tag{1}$$

$$\mathbf{n}^* = \begin{bmatrix} \nabla \mathbf{d} & 2\frac{1}{\text{width}} \end{bmatrix}^T = \begin{bmatrix} \frac{\partial \mathbf{d}}{\partial x} & \frac{\partial \mathbf{d}}{\partial y} & 2\frac{1}{\text{width}} \end{bmatrix}^T, \tag{2}$$

The normal $\mathbf{n}^*$ is derived from the depth map using gradients along horizontal (x) and vertical (y) directions. The z component can be considered a strength factor which is derived from the image width. The total loss is a weighted combination of the three losses: $\mathcal{L}_2^{\text{depth}} + \mathcal{L}_{angular}^{\text{normals}} + 0.5 \times \mathcal{L}_{consistency}^{\text{normals/depth}}$.

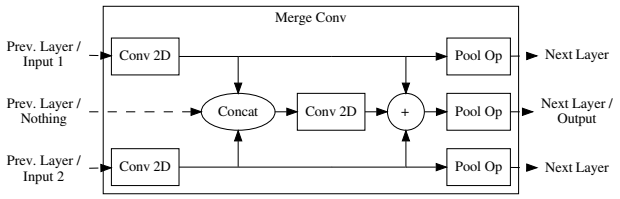

Figure 3: **Merge convolutions.** The merge convolution provides separate pathways for the two-shot inputs and merges the information in a third path.

3.2. Shape Guided Illumination Estimation

To guide SVBRDF predictions, we also estimate the environment illumination. Hereby, the BRDF prediction can consider environment light and reduce additional highlights as well as improve the albedo colors and intensities. The illumination is represented with 24 spherical Gaussians (SG), where each SG is defined by amplitude, axis, and sharpness. However, we only estimate the amplitude and set the axis and sharpness to cover a unit sphere. The estimation thus only estimates the amplitudes of the SG resulting in 24 RGB values. As the environment illumination can reach very high values and the flash and no-flash input images are in LDR, SG amplitudes are constrained to values between 0 and 2. Refer to the supplementary for environment map samples and their SG representations.

We use a small convolutional encoder network followed by fully-connected layers for illumination estimation. The network receives two-shot images, object mask, and the previously predicted depth and normals as input. As illumination is reflected on the surface towards the viewer, the previously estimated shape information helps in better illumination estimations. To train the illumination network, we use the $\mathcal{L}_2$ distance between predicted and ground-truth SGs as the loss function.

3.3. Guided SVBRDF Estimation

SVBRDF estimation becomes a less ambiguous task when conditioned on known object shape and environment illumination. Thus, together with two-shot images, the previously estimated depth, normals, and illumination are used as input to the SVBRDF network to predict diffuse albedo and specular color as well as surface roughness at each pixel. Following recent work on BRDF estimation [31, 34, 14], the U-net architecture [47] is used in our SVBRDF network.

Differentiable Rendering We develop a differentiable rendering module to re-render the object flash image from the estimated depth, normals, illumination, and SVBRDF. At each surface point, the renderer evaluates the direct light from the flash-light source and the estimated environment illumination and integrates it with the BRDF to compute

the reflected light [22]. Fast evaluation of the environment illumination is achieved by representing the illumination as well as the BRDF model as spherical Gaussians (SG) [59]. The product of two SGs is an SG, and the integral of an SG has a closed-form solution that is inexpensive to compute.

Loss Functions for SVBRDF Network The SVBRDF network is trained using a combination of different loss terms: the mean absolute error (MAE) between GT and the predicted SVBRDF parameters as well as a loss between a synthetic direct illumination only flash GT image and re-rendered direct illumination flash image. The rendering loss is back-propagated through the differentiable renderer to update the SVBRDF network. As rendering can result in large values from specular highlights, the MAE loss is calculated on $\log(1+x)$, where x refers to the direct light only synthetic input and the re-rendered image.

3.4. Joint Shape and SVBRDF Refinement

In our cascaded network, we use the estimated depth to guide the SVBRDF prediction. Likewise, one can obtain better depth prediction with known SVBRDF. We jointly optimize depth, normals, and SVBRDF using a separate refinement network. For this refinement, all the earlier predictions along with the residual loss image between the re-rendered previous result and the input flash image are used. The network architecture is a small CNN encoder and decoder of 3 steps, each with 4 ResNet blocks [20] in-between. The loss function is an MAE loss between the predicted parameter maps and ground truth ones.

3.5. Implementation

The cascaded networks along with the differentiable renderer are implemented in Tensorflow [1]. The overall pipeline consists of 4 networks, as illustrated in Fig. 2.

Runtime Each of the networks is relatively small, and the overall pipeline takes 700 ms, including rendering for inference on a 256×256 image on an Nvidia 1080 TI GPU. On a Google Pixel 4 mobile device, the evaluation takes roughly 6 seconds. The rendering step takes about 220ms on a single-threaded desktop CPU (AMD Ryzen 7 1700) and similar speeds on Google Pixel 4. Refer to the supplementary for further runtime analysis.

Training All the networks are trained for 200 epochs with 1500 steps per epoch using the ADAM optimizer [24] with a learning rate of $2\mathrm{e}{-4}$ at the beginning, which is reduced by half after 100 epochs. The networks are trained sequentially as each network in the cascade uses the result of earlier networks as input.

Mobile Application for Scene Capture and Inference In addition to producing better results, another major advantage of the cascaded network design compared to a single joint network is that each of the sub-networks is small, and the overall network can fit on mobile hardware. We convert the network models to Tensorflow Lite that runs on mobile hardware and develop a highly practical android application that can successively capture two-shot flash and no-flash images and runs the cascaded network to estimate SVBRDF and shape. We use on-device GrabCut [48] to obtain the object mask. In Fig. 8 a prediction from the mobile application is shown. Refer to supplementary for more details on the mobile application and further predictions.

4. Large-scale SVBRDF & Shape Dataset

It is very time consuming and expensive to scan SVBRF of real-world objects. Since we rely on deep learning techniques for SVBRDF and shape estimation, vast amounts of data are needed for network supervision. We create a large-scale synthetic dataset with realistic SVBRDF materials.

High-quality Material Collection We gather a collection of publicly available human-authored, high-quality SVBRDF maps from various online sources [44, 13, 64, 10, 54, 57]. The parameterization of these collected SVBRDF maps is for the Cook-Torrence model [12]. In total, the collection consists of 1125 high-resolution SVBRDF maps. To further increase the material pool, we randomly resize and take 768×768 crops of these material maps. We additionally apply random overlays together with simple contrast, hue, and brightness changes. The final material pool contains 11,250 material maps. Sample material maps are shown in Fig. 4.

Domain Randomized Object Shapes One option for generating 3D objects is to gather realistic object meshes and apply materials to those. However, it is challenging to collect large-scale object mesh data covering a wide range of object categories. Moreover, mapping the object meshes to the corresponding materials (*e.g.*, using ceramic materials for teapots) would result in a small dataset, and thus, applying random materials to object meshes is a reasonable strategy. We notice that applying random material maps to complex-shaped object meshes would result in distorted texture or tiling artifacts. Because of these numerous challenges, we choose to randomize object shapes to synthesize large-scale data. Following Xu *et al*. [62], a randomly chosen material is applied to 9 different shape primitives such as spheres, cones, cylinders, tori, etc. We randomly choose 6 to 7 material-mapped primitive shapes and place them randomly to assemble a scene. Sample object shape primitives are shown in Fig. 4. This strategy is similar to domain randomization [55] (DR) that is shown to be useful in high-level semantic tasks such as object detection [56]. Here, we demonstrate the use of DR for the low-level yet complex task of SVBRDF and shape estimation. For simplicity, we refer to our material-mapped and geometry randomized object shapes as DR objects. Fig. 4 shows sample primitive shapes, materials and resulting DR objects with GT shape and SVBRDF parameters.

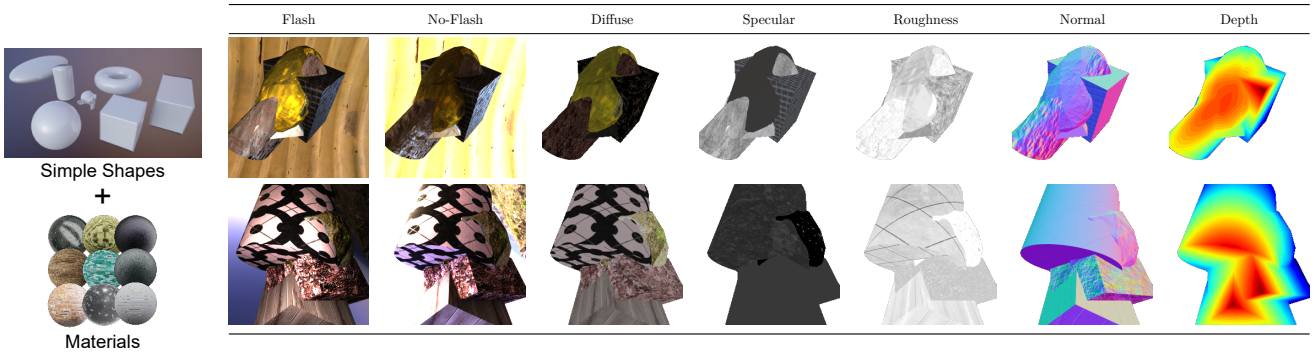

Figure 4: **Large-scale Synthetic Dataset.** (Left) Samples of primitive shapes and materials used for the dataset creation, (Right) The visualization of two examples with various properties.

HDR Illumination For environment illumination, we collect 285 high-dynamic-range (HDR) illumination maps from [63]. These maps are images in latitude-longitude format, which are wrapped on the inside of a sphere, which acts as a light source for the DR object.

Rendering We use the Mitsuba [21] renderer to create two-shot flash and no-flash images of a DR object illuminated with a randomly chosen illumination. In total, the DR dataset contains 100K generated scenes. Note that each DR object consists of differently sampled primitive shapes, and the distance of the closest surface from the camera varies across different DR objects. This setup mimics the real-world capture setting where the object distance to the camera varies. For the no-flash image rendering, the camera position is slightly shifted to mimic the camera shake in a mobile scene capture.

In addition to the two-shot flash and no-flash images, we also render another flash image that only has direct illumination. This direct illumination flash image is used to additionally supervise the SVBRDF network after differentiable rendering (Sec. 3.3). This direct illumination only image is solely used for training supervision and is not required for inference. Besides, we render GT depth, normals, diffuse albedo, specular albedo, and roughness maps, using Mitsuba [21], that are used for direct network supervision. Fig. 4 shows samples from this dataset with more in the supplementary, which also provides additional details on the rendering setup.

5. Experiments

We evaluate our approach on both synthetic and real datasets and compared with several baseline techniques. In this section, we present both quantitative and qualitative results and refer to the supplementary materials for further visual results and comparisons.

Test datasets We quantitatively validate the proposed method on synthetic data with realistic object shapes and SVBRDF and also qualitatively on a real-world two-shot image dataset [5]. Images of both of these datasets are unseen during network training. For synthetic test data, we collected 20 freely available, fully textured 3D objects with realistic shapes and materials [11]. These objects are rendered using the Mitsuba renderer [21] with unseen HDR illumination maps. Fig. 5 and 6 show samples of two-shot input images of our synthetic test dataset.

For real-world evaluation, we use two-shot images from the recent 'flash and ambient illuminations dataset' from [5]. We have created foreground object masks on several samples from the 'Objects' and 'Toys' category, as these fit the single object assumption. This dataset does not contain ground truth BRDF parameters, but the visual quality can be inspected on the estimations and also on re-renderings with different camera views and illuminations.

Metrics To evaluate the quality of the shape and SVBRDF predictions, we mainly use metrics that directly compare the ground truth (GT) and predictions. For the depth and normal estimations, a Mean Square Error (MSE) is a fitting candidate. To enable comparisons with methods that predict relative depths, we employ a Scale-Shift Invariant Metric as in [25]. Refer to the supplementary for details. For SVBRDFs, there exists no clear metric which aligns with human perception of materials. Following previous works, we also use the MSE metric on SVBRDF prediction maps.

5.1. Ablation Study

Within our framework, we empirically evaluate different choices we make in our network design.

Cascade vs. Joint Network We compare our cascaded network with a single large joint network that estimates all the shape and SVBRDF parameters together. For a fair comparison, we design a joint (JN) that has a comparable number of network parameters as our cascaded network (CN) ('Ours-CN' vs. 'Ours-JN'). The JN follows the U-Net [47] architecture. Table 1 shows the quantitative comparisons between them. Results indicate that the CN consistently

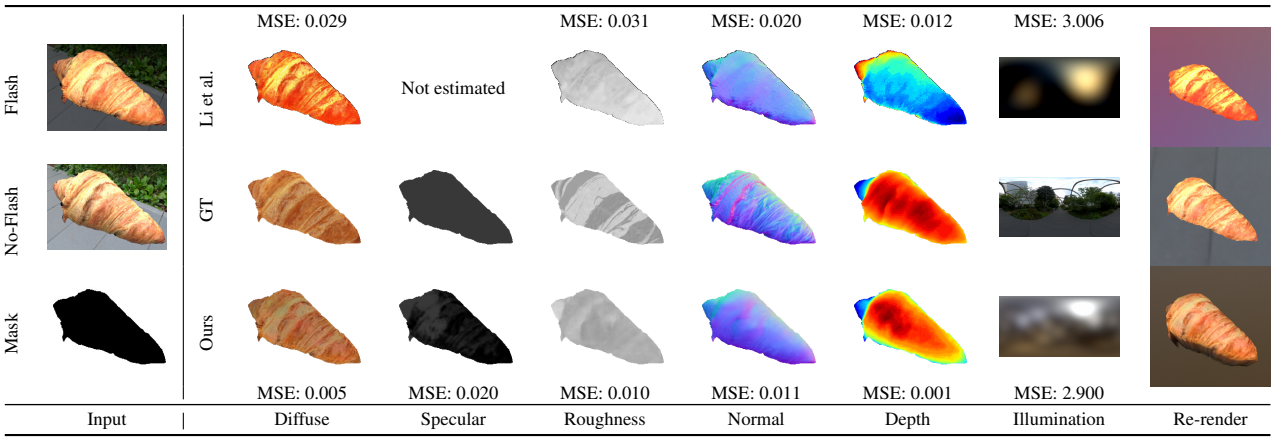

Figure 5: **Comparison with Li *et al*. [35].** Ours estimates the diffuse, depth and normal more accurately in particular.

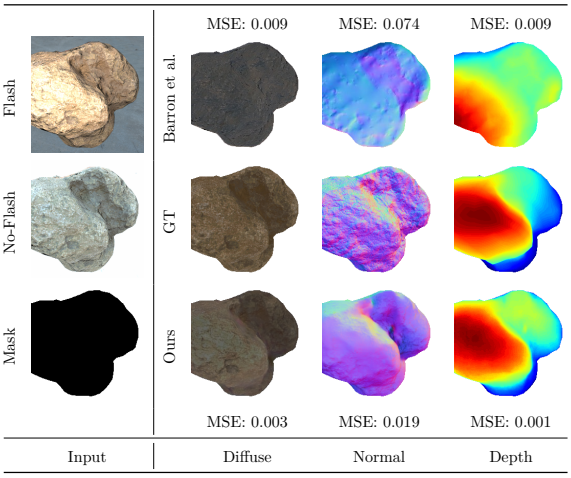

Figure 6: **Comparison with Barron *et al*. [7] (SIRFS).** Barron *et al*. does not estimate specular and roughness parameters.

Method	Diffuse	Specular	Roughness	Normal	Depth
MiDaS [25]	NA	NA	NA	NA	[0.006]
SIRFS [7]	[0.033]	NA	NA	0.089	[0.021]
RAFII [42]	**[0.018]**	NA	NA	NA	NA
Li *et al*. [35]	0.160/[0.019]	NA	0.072	0.034	[0.024]
Ours-JN	0.065/[0.022]	0.053	0.064	0.025	[0.005]
Ours-CN	**0.060/[0.018]**	**0.047**	**0.061**	**0.021**	**[0.004]**

Table 1: **State-of-the-art comparison.** The Mean Square Error (MSE) on a sample dataset of 20 unseen objects. Scale and shift invariant metric in [.] where it applies. For the diffuse color this metric is only scale invariant.

outperforms JN on both SVBRDF and shape estimations, by a significant margin. This empirically underlines the usefulness of our guided stage-wise estimation and joint refinement compared to using a single large network for joint SVBRDF and shape estimation.

Merge vs. Standard Convolutions for Shape Estimation Another technical innovation in this work is the use of MergeConv blocks (Sec. 3.1) in the shape estimation network instead of standard convolution. Overall the depth estimation error decreased from a MSE of 0.021 to 0.016 and the normal MSE from 0.026 to 0.021.

5.2. Comparisons with state-of-the-art

As per our knowledge, we are the first work that uses two-shot images as input and does complete SVBRDF estimation, including specular color and shape estimation for objects. Most existing closely related techniques usually use a single flash image as input and either work only on flat surfaces [14, 15, 34, 31], or do not estimate the specular color [35]. Although our approach features a unique setting, we perform the comparisons with SIRFS [7], Li *et al*. [35], and RAFII [42] on SVBRDF and shape estimation. SIRFS [7] uses a no-flash single image as input and predicts diffuse albedo, shading, and shape using an optimization-based approach. RAFII [42] uses a single non-flash image to perform the intrinsic decomposition. Visual results are shown in the supplementary. Based on a single flash image Li *et al*. [35] is a recent deep learning approach that predicts diffuse albedo, roughness, normal, and depth maps.

Quantitative results on the 20 objects synthetic test dataset shown in Table 1 demonstrate the superior performance of our approach (Ours-CascadeNet) compared to both SIRFS [7] and Li *et al*. [35]. Since SIRFS predicts diffuse albedo only up to a scale factor, we also report scale-invariant MSE scores on diffuse albedo. Fig. 5 shows a visual comparison with Li *et al*. [35]. Our estimations are also visually closer to GT. Especially, we can observe clear visual differences in predicted diffuse albedos where the light information is separated much better in our result. Furthermore, the general shape of the object in the normal map of our method follows the contour of the croissant, while the method of Li *et al*. predicts a mostly flat shape. The details

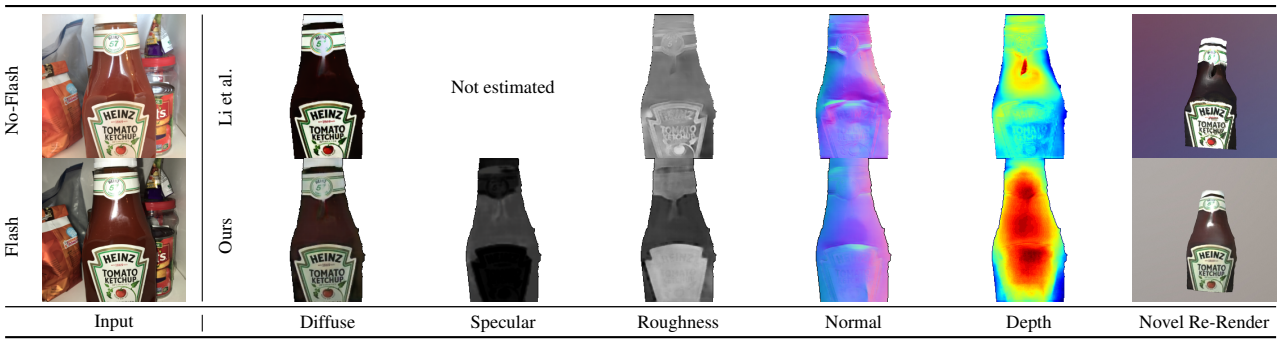

Figure 7: **Real-world comparison.** Comparison with Li *et al*. [35] on a real-world sample from [5].

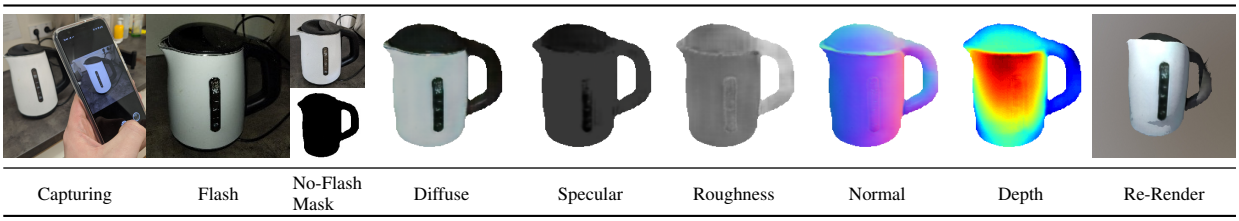

Figure 8: **Mobile capture and inference.** A result from our mobile application that does two-shot image capture followed by SVBRDF and shape estimation.

in the roughness and normal map, on the other hand, are not perfectly predicted by neither method.

Fig. 6 shows a visual comparison with SIRFS, where we again observe our method predictions to be closer to GT. Here, the improvements in the diffuse and normal map are apparent. The SIRFS method fails in this example to separate shape from shading.

A visual comparison between Li *et al*. [35] on a real-world example from Yagiz *et al*. [5] is shown in Fig. 7. Our method seems to capture the object color as well as the shape better. The shape from Li *et al*. is predicted as a nearly flat surface. This is apparent in the novel re-rendering. Our predicted normal map is also smoother with fewer artifacts and follows the bottle shape closely.

For evaluating depth prediction, we compare our depth estimates against those from a new state-of-the-art monocular depth network of MiDaS [25]. MiDaS is trained with several existing depth datasets and is quiet robust to different scene types. MiDaS [25] predicts the relative depth, and for comparisons, a scale-shift invariant MSE metric is used. Table 1 shows the results indicating better depth estimations using our approach. We present qualitative results in the supplementary.

Mobile capture and inference To further showcase our real-world performance, Fig. 8 presents an example captured with our mobile application. As seen, most parameters are plausible. The lid on top of the electric kettle is, however, estimated slightly too far away in the depth map. This can be attributed to the 'deep is dark' ambiguity. Here, we want to point out that there is an additional challenge of an unknown mobile camera capture pipeline. A RAW image capture would avoid most of the unknown image preprocessing in modern cameras.

6. Conclusion

We proposed a novel cascaded network design coupled with guided prediction networks for SVBRDF and shape estimation from two-shot images. Our key insight is that the separation of tasks and stage-wise prediction can lead to significantly better results compared to joint estimation with a single large network. We use a two-shot capture setting, which is practical and helps in estimating higher quality SVBRDF and shape compared to existing works. All of our image capture, network inference, and rendering can be easily implemented on mobile hardware. Another key contribution is the creation of large-scale synthetic training data with domain-randomized geometry and carefully collected materials. We show that networks trained on this data can generalize well to real-world objects. In the future, we would like to tackle the SVBRDF estimation of more complex mirror-like objects by incorporating reflection removal techniques and anisotropic BRDF models.

Acknowledgement This work was partly funded by the Deutsche Forschungsgemeinschaft (German Research Foundation) - Projektnummer 276693517 - SFB 1233. We thank Ben Eckart for his help in the supplementary video.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 5

[2] M. Aittala, T. Aila, and J. Lehtinen. Reflectance modeling by neural texture synthesis. In *ACM Transactions on Graphics (ToG)*, 2018. 2

[3] M. Aittala, T. Weyrich, and J. Lehtinen. Practical svbrdf capture in the frequency domain. In *ACM Transactions on Graphics (SIGGRAPH)*, 2013. 2

[4] M. Aittala, T. Weyrich, and J. Lehtinen. Two-shot SVBRDF capture for stationary materials. In *ACM Transactions on Graphics (ToG)*, 2015. 1, 2

[5] Y. Aksoy, C. Kim, P. Kellnhofer, S. Paris, M. Elgharib, M. Pollefeys, and W. Matusik. A dataset of flash and ambient illumination pairs from the crowd. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 6, 8

[6] N. G. Alldrin, T. Zickler, and D. Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[7] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2015. 2, 3, 7

[8] H. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, 1978. 3

[9] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. In *ACM Transactions on Graphics (SIGGRAPH)*, 2014. 3

[10] Brian. freepbr, 2019. https://freepbr.com. 5

[11] CgTrader. Free 3d models, 2019. www.cgtrader.com. 6

[12] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1982. 2, 3, 5

[13] L. Demes. Cc0 textures, 2019. https://cc0textures.com/. 5

[14] V. Deschaintre, M. Aitalla, F. Durand, G. Drettakis, and A. Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. In *ACM Transactions on Graphics (ToG)*, 2018. 2, 4, 7

[15] V. Deschaintre, M. Aitalla, F. Durand, G. Drettakis, and A. Bousseau. Flexible SVBRDF capture with a multi-image deep network. In *Eurographics Symposium on Rendering*, 2019. 7

[16] Y. Dong, J. Wang, X. Tong, J. Snyder, Y. Lan, M. Ben-Ezra, and B. Guo. Manifold bootstrapping for svbrdf capture. In *ACM Transactions on Graphics (SIGGRAPH)*, 2010. 2

[17] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[18] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[19] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009. 2

[20] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016. 4, 5

[21] W. Jakob. Mitsuba - physically based renderer, 2018. https://www.mitsuba-renderer.org/. 6

[22] J. T. Kajiya. The rendering equation. In *ACM Transactions on Graphics (SIGGRAPH)*, 1986. 5

[23] K. Kim, J. Gu, S. Tyree, P. Molchanov, M. Nießner, and J. Kautz. A lightweight approach for on-the-fly reflectance estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[25] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *ArXiv e-prints*, 2019. 2, 3, 6, 7, 8

[26] H. Lensch, J. Kautz, M. Gosele, and H.-P. Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Conference on Rendering*, 2001. 2, 3

[27] H. P. Lensch, J. Lang, M. S. Asla, and H. Seidel. Planned sampling of spatially varying brdfs. In *Computer graphics forum*, 2003. 2

[28] L. Lettry, K. Vanhoey, and L. Van Gool. DARN: a deep adversarial residual network for intrinsic image decomposition. In *IEEE International Conference on Computer Vision (ICCV)*, 2018. 3

[29] M. Li et al. Deep spherical Gaussian illumination estimation for indoor scene. In *ACM Multimedia Asia Conference (MM Asia)*, 2019. 3

[30] R. Li, K. Xian, C. Shen, Z. Cao, H. Lu, and L. Hang. Deep attention-based classification network for robust depth prediction. In *Asian Conference on Computer Vision (ACCV)*, 2019. 3

[31] X. Li, Y. Dong, P. Peers, and X. Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. In *ACM Transactions on Graphics (ToG)*, 2017. 2, 4, 7

[32] Z. Li and N. Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[33] Z. Li and N. Snavely. Learning intrinsic image decomposition from watching the world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[34] Z. Li, K. Sunkavalli, and M. Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 4, 7

[35] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018. 2, 3, 7, 8

[36] M. Liao, L. Wang, R. Yang, and M. Gong. Light fall-off stereo. In *IEEE International Conference on Computer Vision (ICCV)*, 2007. 3

[37] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2016. 3

[38] R. Maier, K. Kim, D. Cremers, J. Kautz, and M. Nießner. Intrinsic3D: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[39] A. Meka, M. Maximov, M. Zollhoefer, A. Chatterjee, H.-P. Seidel, C. Richardt, and C. Theobalt. Lime: Live intrinsic material estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[40] G. Nam, D. Gutierrez, and M. H. Kim. Practical SVBRDF acquisition of 3d objects with unstructured flash photography. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018. 2, 3

[41] T. Narihira, M. Maire, and S. X. Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[42] T. Nestmeyer and P. V. Gehler. Reflectance adaptive filtering improves intrinsic image estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 7

[43] F. E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied Optics*, 1965. 1

[44] J. Paulo. 3d textures, 2019. https://3dtextures.me/. 5

[45] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *ACM Transactions on Graphics (SIGGRAPH)*, 2001. 1

[46] P. Ren, Y. Dong, S. Lin, X. Tong, and B. Guo. Image based relighting using neural networks. *ACM Transactions on Graphics (ToG)*, 2015. 1

[47] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015. 4, 6

[48] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 2004. 3, 5

[49] A. Roy and S. Todorovic. Monocular Depth Estimation Using Neural Regression Forest. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[50] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[51] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz. Neural inverse rendering of an indoor scene from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[52] J. Shi, Y. Dong, H. Su, and S. X. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[53] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2005. 3

[54] S. Textures. Share textures, 2019. https://sharetextures.com. 5

[55] J. Tobin, R. Fong, A. Ray, J. Schneider, Z. Wojciech, and P. Abeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent RObots and Systems IROS*, 2017. 2, 5

[56] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018. 5

[57] R. Tuytel. Texture haven, 2019. https://texturehaven.com. 5

[58] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. DeMoN: depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[59] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2009. 5

[60] T. Y. Wang et al. Joint material and illumination estimation from photo sets in the wild. In *International Conference on 3D Vision (3DV)*, 2018. 3

[61] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[62] Z. Xu et al. Deep image-based relighting from optimal sparse samples. *TOG*, 2018. 5

[63] G. Zaal. Hdri haven, 2019. https://hdrihaven.com/. 6

[64] D. Zgraggen. cgbookcase, 2019. https://cgbookcase.com/. 5

[65] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1999. 2

[66] T. Zhou, P. Krähenbühl, and A. A. Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3