Deep Inverse Rendering for High-resolution SVBRDF Estimation from an Arbitrary Number of Images

DUAN GAO, Tsinghua University and Microsoft Research Asia
XIAO LI, University of Science and Technology of China and Microsoft Research Asia
YUE DONG, Microsoft Research Asia
PIETER PEERS, College of William & Mary
KUN XU, Tsinghua University
XIN TONG, Microsoft Research Asia

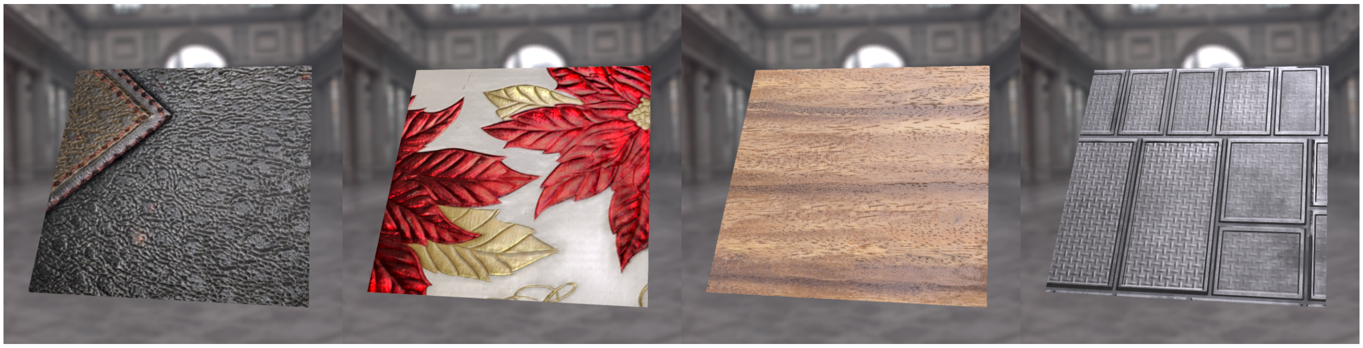

Fig. 1. Visualizations under natural lighting of four captured $1k$ resolution SVBRDFs estimated using our deep inverse rendering framework. The leather material (left) is reconstructed from just 2 input photographs captured with a mobile phone camera and flash, while the other materials are recovered from 20 input photographs.

In this paper we present a unified deep inverse rendering framework for estimating the spatially-varying appearance properties of a planar exemplar from an arbitrary number of input photographs, ranging from just a single photograph to many photographs. The precision of the estimated appearance scales from plausible when the input photographs fails to capture all the reflectance information, to accurate for large input sets. A key distinguishing feature of our framework is that it directly optimizes for the appearance parameters in a latent embedded space of spatially-varying appearance, such that no handcrafted heuristics are needed to regularize the optimization. This latent embedding is learned through a fully convolutional auto-encoder that has been designed to regularize the optimization. Our framework not only supports an arbitrary number of input photographs, but also at high resolution. We demonstrate and evaluate our deep inverse rendering solution on a wide variety of publicly available datasets.

CCS Concepts: • **Computing methodologies → Reflectance modeling**; *Image processing*.

Additional Key Words and Phrases: Material Capture, SVBRDF, Deep Learning, Auto-encoder.

ACM Reference Format:
Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. 2019. Deep Inverse Rendering for High-resolution SVBRDF Estimation from an Arbitrary Number of Images. *ACM Trans. Graph.* 38, 4, Article 134 (July 2019), 15 pages. https://doi.org/10.1145/3306346.3323042

1 INTRODUCTION

Estimating the surface reflectance properties of a spatially-varying material is a challenging problem. Methods based on inverse rendering (e.g., [Dong et al. 2014; Hui et al. 2017]) can obtain accurate estimates for a sufficiently large number of input photographs. However, if the number of photographs is too low, such inverse rendering methods fail to produce plausible results. Recently, a number of techniques have been presented that, leveraging recent advances in deep learning, focus on achieving plausible results from just a single image [Deschaintre et al. 2018; Li et al. 2017, 2018a,b; Ye et al. 2018]. However, these methods fail to reproduce reflectance features that are ambiguous and/or not visible in the single input photograph. For example, specular features that are not excited by the incident lighting in the input photograph can only be inserted based on learned heuristics. Adding one or more photographs that provide

The first two authors contributed equally to this paper. Part of this work was performed while Pieter Peers visited Microsoft Research Asia.

Authors' addresses: D. Gao and K. Xu, Department of Computer Science and Technology, Tsinghua University, Beijing, China;, emails:gao-d17@mails.tsinghua.edu.cn, xukun@tsinghua.edu.cn; X. Li, University of Science and Technology of China, Hefei, China; email:pableetoli@gmail.com; Y. Dong and X. Tong, Microsoft Research Asia, Beijing, China; emails:yuedong, xtong@microsoft.com; P. Peers, Computer Science Department, College of William & Mary, Williamsburg, VA, 23187, USA; email:ppeers@siggraph.org.

0730-0301/2019/7-ART134 $15.00
https://doi.org/10.1145/3306346.3323042

additional cues on the missing or ambiguous features could greatly improve the appearance reconstruction. However, it is unclear how such deep learning based methods can be extended to multiple input images.

In this paper we propose a unified framework for estimating high resolution surface reflectance properties of a spatially-varying planar material sample from an arbitrary number of photographs (Figure 1). The precision of the estimated spatially-varying bidirectional reflectance distribution functions (SVBRDFs) gracefully scales from *"plausible"* approximations when the input images fail to reveal all the reflectance details (e.g., for a single input photograph) to *"accurate"* reproductions for sufficiently large sets of input images. To achieve this goal, our method, named *deep inverse rendering*, combines deep learning and inverse rendering in a flexible and easy to implement framework that performs the inverse rendering optimization in a learned latent space characterized by a fully convolutional auto-encoder [Hinton and Salakhutdinov 2006] which models the space of SVBRDFs. The optimization itself is driven by the rendering error on the reflectance property maps corresponding to the latent vector which is updated via backpropagation through a differential rendering layer. Optimizing the latent vector, instead of the reflectance property maps, constrains the solution to lie in the modeled SVBRDF space. However, to facilitate optimization, careful design of the latent space is needed.

Batch normalization [Ioffe and Szegedy 2015] is the de-facto standard strategy for regularizing convolutional neural networks, including auto-encoders. However, the goal of our SVBRDF auto-encoder differs from traditional convolutional auto-encoders which are trained to minimize the reproduction differences between the input and output. Instead, we desire an SVBRDF auto-encoder that defines a latent space suitable for inverse rendering optimization. We argue and show that batch normalization aversely affects the quality of gradients obtained by backpropagating the rendering loss through the decoder; we therefore omit batch normalization on the decoder. Furthermore, we further improve the robustness of the optimization in the latent space by introducing an additional smoothness loss during the auto-encoder training such that small changes in the latent vector correlate to small changes in the decoded SVBRDF.

While encoder batch normalization and the smoothness constraint improve robustness with respect to optimization, our deep inverse rendering still requires a plausible SVBRDF as a starting point, especially in the case of a few input photographs, since the auto-encoder only models the space of SVBRDFs, not its plausibility. We therefore opt to use the plausible SVBRDF estimates provided by prior single image convolutional neural network solutions [Deschaintre et al. 2018; Li et al. 2018a] as a starting point. In Section 5 we provide a probabilistic interpretation on the role of the starting point.

An advantage of using a fully convolutional auto-encoder for modeling the embedded latent space is that we can easily support high resolution SVBRDFs by simply expanding the resolution of the latent feature maps; this does not require any retraining of the auto-encoder.

We demonstrate our deep inverse rendering solution on a wide variety of SVBRDFs from different publicly available SVBRDF datasets, as well as on captured photographs of spatially-varying materials. Furthermore, we show that, for single input photographs, our solution improves the quality of the estimated SVBRDFs compared to prior learning-based approaches [Deschaintre et al. 2018; Li et al. 2018a].

In summary, our deep inverse rendering solution for recovering SVBRDFs:

(1) can operate with an arbitrary number of input photographs, ranging from as few as one to many;
(2) is not limited to a fixed input and output resolution; and
(3) improves the quality of SVBRDFs estimated from single photograph inputs compared to prior work, especially when the target material falls outside the training dataset.

2 RELATED WORK

The ubiquitous availability of cheap digital cameras presents an opportunity for non-expert users to employ image-driven modeling tools formerly restricted to specialized labs. In the past decade, several advances have been presented for bringing appearance modeling of spatially-varying materials to the masses. A key prerequisite for such appearance modeling methods is that they need to be robust to suboptimal choices in the acquisition parameters. Therefore, we focus this overview of related work on methods that endeavor to simplify the acquisition process and that rely on a lightweight acquisition setup using consumer cameras. We refer to the excellent overviews of Dorsey et al. [2008] and Weinmann and Klein [2015] for a detailed overview of general appearance modeling techniques.

Multi-Image Heuristics-based Appearance Modeling. A first class of methods aims to model the appearance as accurately as possible, while emphasizing ease of acquisition and simplicity of the acquisition process.

Riviere et al. [2016] record a video of a spatially-varying sample using a mobile phone and lit by a flash light. They fit surface normals and a BRDF per surface point using handcrafted heuristics to identify specular and diffuse reflections. Hui et al. [2017] also record a video using a cellphone camera and flash. However, unlike Riviere et al., they *"scan"* the surface. Hui et al. iteratively fit surface normals and a dictionary-based BRDF model assuming sparsity. Palma et al. [2012] recover the spatially-varying appearance from a video sequence of an object under fixed natural lighting. Similar to Riviere et al., a heuristic is used to separate and fit the diffuse and specular reflectance. Dong et al. [2014] estimate surface normals and reflectance properties for each surface point from a video of a rotating object under unknown natural lighting by exploiting the sparsity of strong edges in the incident lighting. This was further extended Xia et al. [2016] to also recover the shape of the object.

Despite using multiple observations of the scene, these methods still require regularization due to the limitations of the light-weight acquisition process, often in the form of handcrafted heuristics or by assuming sparsity in some domain. Furthermore, these methods have a hard lower bound on the number of input images for which a meaningful reflectance estimate can be obtained. In contrast, our method aims to produce valid SVBRDFs starting from any arbitrary number of input photographs.

Single/Few Image Reflectance Modeling. Another class of methods is designed to reduce the number of input images to a minimum while recovering a plausible estimate of the material properties.

Aittala et al. [2015] recover the reflectance properties of texture-like materials from just two photographs (one with flash and one without) by exploiting that each local region is statistically similar (i.e., stationary), while receiving lighting from different directions under the flash lighting. Xu et al. [2016] also exploit spatial relations and recover material properties from just two photographs from a near-field perspective camera. While originally designed for homogeneous materials, it can also be applied to piece-wise spatially-varying materials. Zhou et al. [2016] present a general framework that also exploits spatial relations and sparseness in basis BRDF representations. Their method can recover the SVBRDF (excluding surface normals) ranging from just one image assuming a piece-wise spatially-varying material, to more detailed spatial variations from multiple input images. The above methods all make strong assumptions on some form of spatial sparseness of the material. In contrast, our method relies on learned features that encode spatial relations (not necessarily sparse).

Learning-based Appearance Modeling. Li et al. [2017] proposed a novel self-augmentation training strategy that only requires a small labeled training set of measured SVBRDFs in conjunction with a large unlabeled set of regular photographs of spatially-varying materials for learning a material-class specific convolutional neural network that can infer the reflectance properties of a planar material sample from a single photograph under unknown natural lighting. Ye et al. [2018] improve on this, and remove the need for labeled training data altogether.

Deschaintre et al. [2018] and Li et al. [2018a] estimate reflectance properties using a convolutional neural network from a single photograph lit by flash lighting. Both solutions differ in their network architecture. Deschaintre et al. consider both global information provided by local lighting as well as texture detail. Li et al. add a densely connected conditional random field post-processing step to further enhance the estimated reflectance parameters. In this paper, we will use the single photograph estimation networks of Deschaintre et al. [2018] and of Li et al. [2018a] to bootstrap our deep inverse rendering optimization.

Li et al. [2018b] introduced a deep learning based solution for recovering shape and spatially-varying reflectance from a single image. Key to their method is a cascading network structure to iteratively refine the solution, and a rendering layer that predicts the next *"bounce"* of indirect lighting.

While powerful, the above deep learning solutions are designed for a single input photograph, and it is not straightforward to extend them to handle multiple input images. One strategy would be to fix the number of input photographs beforehand, as implemented by Xu et al. [2018] in the context of the related problem of image-based relighting. A more flexible solution was presented by Kim et al. [2017] who reconstruct a *homogeneous* BRDF from multiview observations using a multi-level fully connected deep network design. Key to their method is the inclusion of a novel *"moment pooling layer"* that aggregates the features from multiple observations. However, their solution is limited to homogeneous BRDFs. All of the above deep learning based appearance modeling methods use a (convolutional) neural network to directly infer the reflectance properties from the input photographs. In contrast, we follow a more traditional inverse rendering approach guided by learned features. While computationally more expensive during estimation, it also inherits the flexibility of optimization-based solutions with respect to the number of input photographs.

Closest related to deep inverse rendering is the method of Aittala et al. [2016] who estimate spatially-varying surface normals and reflectance properties of a planar stationary material sample from a single photograph lit by flash lighting. They perform an inverse rendering approach driven by a powerful learned texture descriptor [Gatys et al. 2015] to softly compare the predictions. Similar to Aittala et al., we also combine deep learning and inverse rendering. However, whereas Aittala et al. project *rendered predictions (images)* into a general texture-feature space, we directly optimize the features of the *reflectance parameter maps* in a learned embedding of SVBRDFs.

Optimizing with Auto-encoders. Kang et al. [2018] learn illumination patterns for efficient capture of surface reflectance properties. Key to their method is an asymmetric auto-encoder that that features a linear non-negative encoder that corresponds to the acquisition lighting, and a non-linear decoder that maps the measurements to reflectance information, which in turn is fitted to an analytical BRDF model. From the perspective of inverse rendering, Kang et al.'s approach can be seen as using deep learning to provide a starting point for the optimization. We do not only rely on deep learning to provide a starting point, but also to drive the optimization.

Choi et al. [2017] model the space of multispectral images using an auto-encoder to aid in reconstructing multispectral images from compressive hyperspectral measurements. Similar to us they solve the reconstruction problem by directly optimizing in the latent space. However, to regularize the reconstruction a cycle-loss term is introduced (enforcing that the encoding of the decoded latent variable is valid) as well as a total variation regularizer that favors sparse image gradients. Calian et al. [2018] estimate a light probe from a single image of a face. Similar to Choi et al., Calian et al. do not only use the latent space for representing the target (i.e., environment lighting), but also for modeling an additional optimization regularization term. We take inspiration from these methods and also optimize in the latent space. However, we do not add handcrafted heuristics (such as gradient sparseness) to regularize the optimization, but instead adjust the latent space to be more robust to optimization. Explicitely including a regularization term requires careful balancing of both terms. Tuning this balance is non-trivial and likely dependent on the properties and quantity of input images.

3 OVERVIEW

Preliminaries. Our goal is to estimate the reflectance properties of a spatially-varying material from an arbitrary number of photographs. We assume that the material sample is planar with the exception of small-scale surface details that can be modeled by a normal map. Furthermore, we assume that the surface reflectance at each surface point can be well represented by the Cook-Torrance microfacet BRDF model using the GGX distribution as the microfacet

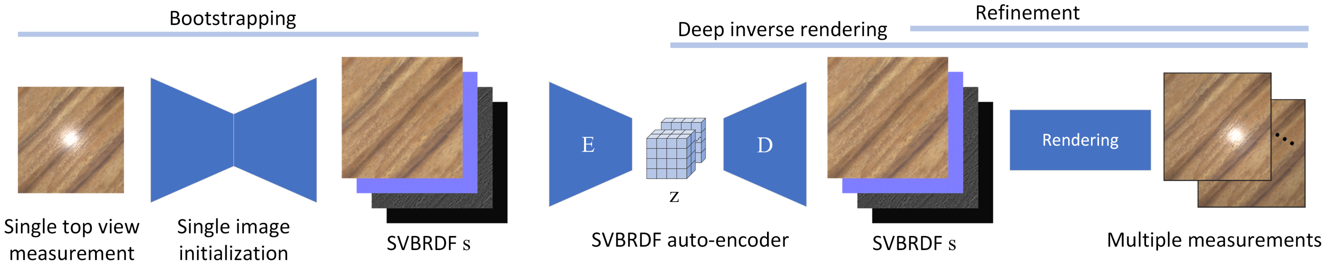

Fig. 2. Overview of the deep inverse rendering framework. The core of our system is an SVBRDF auto-encoder specifically designed for deep inverse rendering. Key to deep inverse rendering is the optimization of the appearance parameters directly in the latent embedded space of spatially-varying appearance, driven *only* by the rendering loss of the decoded SVBRDF to the input photographs (i.e., without any additional handcrafted regularization terms). To bootstrap the optimization, we use an existing single image SVBRDF estimation network [Deschaintre et al. 2018] that estimates the most plausible SVBRDF from a single top view photograph. Finally, we perform a refinement post-processing step to reintroduce fine details not encoded in the latent space.

normal distribution [Walter et al. 2007]. The reflectance of each surface point p is characterized by: a surface normal $n(p)$, diffuse albedo $k_d(p)$, specular albedo $k_s(p)$, and a monochrome specular roughness $\alpha(p)$. Furthermore, we assume that the photographs $\{I_i\}_i$ of the material are lit by a point light source (approximately) co-located with the camera. We impose no restrictions on the camera locations for each photograph although we try to keep the distance to the sample approximately constant. Furthermore, we require that at least one photograph I_0 is captured from approximately normal incidence. For each photograph I_i we assume that the internal and external camera intrinsics C_i are known and that each photograph is radiometrically linearized and rectified to the top view; we will define all material property maps with respect to the top view. Our method supports both high-dynamic range as well as 8-bit (linear) low dynamic range input photographs.

Deep Inverse Rendering. Our deep inverse rendering framework uses the classic inverse rendering approach and formulates the estimation of the reflectance parameters $s = (n, k_d, \alpha, k_s)$ as a minimization that attempts to minimize the differences, according to some loss function $\mathcal{L}(\cdot,\cdot)$, between the photographs $\{I_i\}_i$ and the rendering[1] $R(s, C_i)$ of the reflectance parameters and the camera (and light source) parameters C_i:

$$\operatorname*{argmin}_s \sum_i \mathcal{L}(I_i, R(s, C_i)). \tag{1}$$

For the loss function $\mathcal{L}(\cdot,\cdot)$, we follow Deschaintre et al. [2018] and use the L_1 distance on log-encoded pixels values to reduce the impact of high intensity specular pixel values:

$$\mathcal{L}(x, y) = ||log(x + 0.01) - log(y + 0.01)||_1. \tag{2}$$

However, unlike traditional inverse rendering methods, we do not directly optimize the reflectance parameters s, but instead find a solution z in a latent space:

$$\operatorname*{argmin}_z \sum_i \mathcal{L}(I_i, R(D(z), C_i)). \tag{3}$$

A key difference between Equations (1) and (3) is that the former optimizes the reflectance parameters s per pixel, whereas the latter optimizes the whole image with respect to the latent vector z. The latent space embeds the relevant properties and relations of the space of SVBRDFs and serves to regularize the optimization. In our solution we model this latent space using a fully convolutional auto-encoder that consists of an encoder $E(\cdot)$ that transforms an SVBRDF s to its corresponding latent vector z, and a decoder $D(\cdot)$ that translates the latent vector z to the corresponding SVBRDF s:

$$z = E(s), \tag{4}$$
$$s = D(z). \tag{5}$$

The exact details of the SVBRDF auto-encoder are discussed in Section 4. Figure 2 summarizes our deep inverse rendering framework.

Discussion. Our choice for inverse rendering in a learned space, instead of directly learning a direct inference network, is motivated by practical and theoretical constraints.

On the practical side, there is the challenge of dealing with a variable number of input photographs, each of which might be recorded from different viewpoints. Unless a specially designed setup that repeats the same acquisition pattern is used, the viewpoints and/or lighting directions are unknown before acquisition. This poses a significant challenge for deep learning strategies such as convolutional neural networks used in prior work [Deschaintre et al. 2018; Li et al. 2018a] as these networks require such knowledge at training time. By embedding the deep learning component inside a classic inverse rendering pipeline, we also inherit the flexibility of inverse rendering when it comes to handling a wide variety of input conditions. In addition, it enables our framework to handle acquisition parameters unknown during training. Furthermore, performing inverse rendering in a learned space also avoids the need for fragile handcrafted regularizers to handle underconstrained and ambiguous conditions.

On the theoretical side, from the perspective of appearance modeling, one can view learning an inference network as moving the appearance optimization to a precomputation phase. During training the network is optimized to minimize the error over the *whole* training set. Consequently, it is possible that there is a difference in the accuracy between the different training samples. Furthermore, no guarantee can be made for exemplars outside the training set. Hence, on an individual exemplar basis, the inference network might not reach the best solution. Our deep inverse rendering framework

[1] We clamp the rendered pixel values to [0, 1] if the input photographs are low dynamic range.

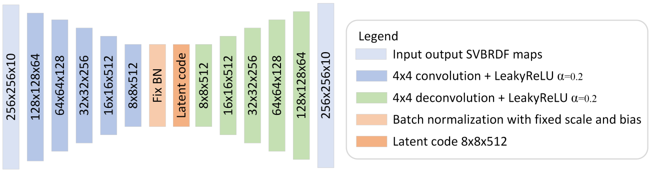

Fig. 3. Auto-encoder network structure.

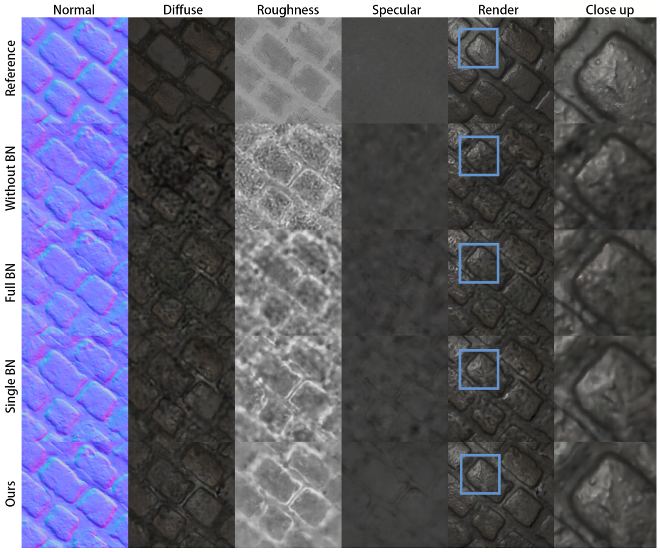

Fig. 4. Deep inverse rendering with different auto-encoder configurations for a single input photograph. The auto-encoder without any batch normalization produces noisy results (2nd row) compared to the reference (top row). In contrast, batch normalization on every layer oversmooths the results (3rd row). Only applying batch normalization on the last encoder layer (just before the latent code) better balances details vs noise (4th row). Single-layer batch normalization with our space smoothness constraint (last row), producing the most plausible result, with clean maps and more accurate details in the renderings (cf. region marked in blue).

does not train an inference network, and only uses the learned space to regularize the optimization. Consequently, it is therefore better equipped to deal with exemplars not seen during training.

4 SVBRDF AUTO-ENCODER FOR DEEP INVERSE RENDERING

Network Architecture. The goal of the auto-encoder is to provide a reduced latent space to perform the optimization in, while at the same time being able to faithfully reproduce the intricate spatial variations observed in spatially-varying materials. As such, we use a standard auto-encoder architecture that takes 256 × 256 reflectance property maps $s = (n, k_d, \alpha, k_s)$ as input and reduces them to 8 × 8 resolution feature maps with a feature length of 512. Figure 3 summarizes the full configuration and convolution filter sizes.

Training Loss Function. We use the same training dataset as Deschaintre et al. [2018]. We explicitely do not use a larger training set to better demonstrate that our deep inverse rendering solution can handle a wider range of spatially-varying materials, as well as to provide a fair comparison to prior work.

The training loss function is a sum of two terms:

$$\mathcal{L}_{train} = \mathcal{L}_{map} + \frac{1}{9}\mathcal{L}_{render}, \tag{6}$$

where $\mathcal{L}_{map}$ is the L_1 loss on the reflectance maps, and $\mathcal{L}_{render}$ is the L_1 log loss on 9 visualizations of the material (similar to the optimization loss (Equation (2)). We follow Deschaintre et al. [2018] and select these 9 images from two sets of distributions. 3 images are selected with independently sampled light and view directions from a cosine weighted distribution over the upper hemisphere. The final 6 images are selected by sampling the lighting directions from the cosine distribution, and setting the view direction to the mirror direction. The viewpoint is randomly selected on the material, and the (log) distance of the camera and lighting are independently sampled from a normal distribution with mean 0.5 and standard deviation 0.75 (assuming a 2 unit square material sample). We will show in Section 7 that the combination of both loss terms yields more stable SVBRDF recovery than either of the terms separately.

Batch Normalization vs. Inverse Rendering. A standard practice for convolutional neural networks is to apply batch normalization [Ioffe and Szegedy 2015] to each convolutional layer with as goal to improve performance and stability in training. However, another feature of batch normalization is that it also acts as a model regularizer. Batch normalization normalizes each feature map for each training batch; a global scale and bias factor are also trained over the model to ensure that the *average* mean and scale remain the same. In effect, batch normalization penalizes features that occur infrequently (e.g., noise). To demonstrate the impact of batch normalization for SVBRDF estimation, we train two auto-encoders: one without any batch normalization, and one with a batch normalization on each convolutional layer. We then use these auto-encoders to optimize an SVBRDF from a single (top view) photograph (Figure 4). The result obtained from the auto-encoder without batch normalization (second row) produces *"noisy"* reflectance maps. In contrast, the result obtained from the auto-encoder with batch normalization (third row) produces *"cleaner"* reflectance maps. However, we also observe that these "clean" reflectance maps loose some sharpness; edge detail and noise are in some sense similar types of features.

To address the loss in edge details, we observe that our goal is different than that of classic auto-encoders. For our application of optimizing latent vectors, we desire a decoder that produces good gradients that capture the details when backpropagated from the rendered images to the latent code. Hence, we posit that applying batch normalization on the decoder overregularizes the auto-encoder; we only want to regularize the latent embedded space to better model the space of SVBRDFs. We demonstrate the validity of our thesis in Figure 4 (fourth row). The single image estimation using an auto-encoder without any batch normalization on the *decoder* exhibits more details than the full batch normalization auto-encoder, but also without the noise introduced by the unregularized auto-encoder. We also experimented with including batch normalization only on the last layer of the encoder (and none on the decoder), and found the results to be similar then when using batch normalization on

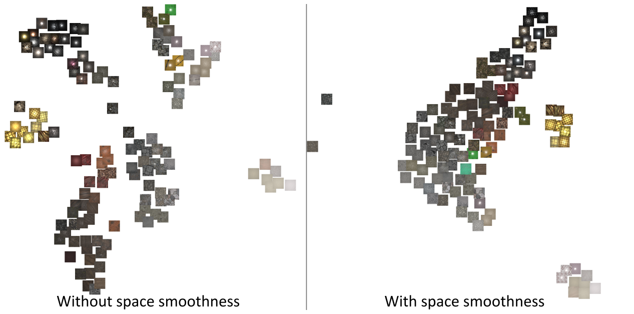

Fig. 5. t-SNE visualizations of the learned latent spaces with and without smoothness loss. With smoothness loss, the resulting latent space is more continuous.

all layers of the encoder. We therefore, opt for using a *single* batch normalization layer (on the last layer of the encoder).

Optimization Space Smoothness. When optimizing the latent code, we rely on backpropagation of gradients to direct the optimization algorithm. These backpropagated gradients are based on the rendered images, and ideally a small change in appearance in the rendered images should result in a small change in latent code. Without regularization during training, there is no guarantee this will be the case. Unfortunately, we do not know at training time what the exact number of input photographs is, nor do we know their respective camera locations, and thus we cannot evaluate the change in appearance during training. Instead, we apply the stronger constraint that a small change in latent code should result in a small change in the SVBRDF (and vice versa). We achieve this by adding a smoothness loss function to the training loss:

$$\mathcal{L}_{smooth} = \lambda_{smooth}||D(z) - D(z + \xi)||_1, \tag{7}$$

where ξ is a random variable drawn from a normal distribution with 0.2 variance and zero mean, and λ_{smooth} is a weight to control the amount of smoothing; we found that $\lambda_{smooth} = 2$ works well in practice. Effectively, this smoothness penalty term attempts to reorganize the latent space such that nearby latent codes have a similar decoded SVBRDF. This is illustrated in Figure 5 via t-SNE visualizations [van der Maaten and Hinton 2008] of the latent spaces trained with and without smoothness loss. The latent space with smoothness loss exhibits a more continuous manifold, which is better suited for interpolation and/or optimization.

A caveat with Equation (7) is that there exists a trivial solution by expanding the scale of latent space; i.e., scaling up the extend of latent space is equivalent to reducing the variance of ξ, and and in the limit case ξ becomes practically equivalent to zero at which point the decoded SVBRDFs are the same. To fix the scale of the latent space, we fix the scale to 1 and bias to 0 in the batch normalization of the latent code. Figure 4 (last row) demonstrates the benefit for deep inverse rendering, yielding a reconstruction with more details than without the smoothness loss during training.

High Resolution Auto-encoder. A fully convolutional auto-encoder has the advantage that we can also encode an input SVBRDF at high resolution. The resulting latent code will be expanded by the same ratio. For example, encoding a 1024 × 1024 SVBRDF yields a 32 × 32 × 512 latent code, or 16 times larger than the 8 × 8 × 512 code of a 256 × 256 SVBRDF. Given this larger latent code, we can decode the SVBRDF again at the input resolution. Hence, our deep inverse rendering framework can easily operate on high resolution SVBRDFs.

5 BOOTSTRAPPING

As in many inverse rendering methods, the initialization of the optimization is essential for obtaining a good result. To better understand the role of the auto-encoder in the optimization, and thus the conditions for the initialization, we express deep inverse rendering in probabilistic terms. Given a set of input photographs $\{I_i\}_i$, we can express inverse rendering as maximizing the conditional probability of an SVBRDF s as: $\text{argmax}_s\, P(s|\{I_i\}_i)$. Using Bayes' theorem and assuming each I_i is independent, we can rewrite this as:

$$P(s|\{I\}_i) = \Pi_i \left(\frac{P(I_i|s)P(s)}{P(I_i)} \right). \tag{8}$$

We assume each image is equally likely, and therefore ignore the image probability $P(I_i)$;

$$P(s|\{I\}_i) = \Pi_i \left(P(I_i|s)P(s) \right). \tag{9}$$

The first probability in Equation (9), $P(I_i|s)$, expresses the probability of an image I_i given the SVBRDF s, and it corresponds to (the corresponding terms in) the fitting loss function. The second probability, $P(s)$, expresses the probability of the SVBRDF s. When the number of input images is large, Equation (9) is dominated (and thus sufficiently constrained) by the rendering loss. Intuitively, many SVBRDFs can potentially explain the input image I_i, and thus its probability will be significant for a large portion of the SVBRDF space. The joint probability of all images is represented by the intersection of these probabilities; the more images there are, the more likely the intersected probability region shrinks in size, and in the limit it will only peak for a single SVBRDF. Consequently, the role of $P(s)$ is less important. However, when the number of input images is low, the conditional probability is significant for a large number of SVBRDFs, and the role of $P(s)$ as a regularizer becomes more important. It is tempting to think that this probability is offered by the auto-encoder. However, this is incorrect. The auto-encoder only provides a latent embedding of the space of SVBRDFs (i.e., a model of the space of SVBRDFs); it does not differentiate between the probability of different embeddable SVBRDFs. However, the smoothness loss (Equation (7)) compels the training to organize the latent space such that nearby latent vectors decode to a similar SVBRDF. Assuming that similar plausible SVBRDFs have a similar probability, we can approximate $P(s)$ locally as a constant:

$$P(D(z)) \approx P(D(z + \xi)). \tag{10}$$

The smaller ξ the more likely that this approximation holds. This suggests that if the starting point z_0 is chosen to fall in the region where the shape of the distribution is dominated by the rendering loss (i.e., $P(s) \approx \textit{constant}$), then we can expect the optimization to converge to a good solution. Unfortunately, this implies we would need to select z_0 close to $\text{argmax}_s\, P(s|\{I_i\}_i)$, which is exactly the probability we want to maximize in the first place. However, the smoothness loss attempts to increase the region (i.e. ξ) for which Equation (10)

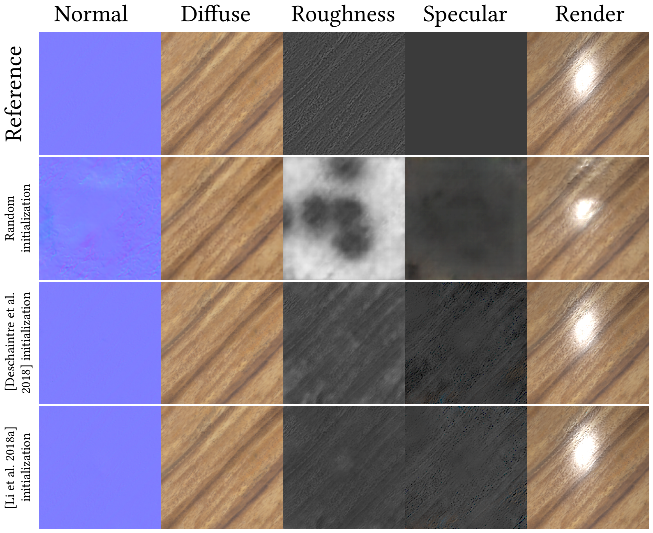
Normal
Diffuse
Roughness
Specular
Render
Reference
Random initialization
[Deschaintre et al. 2018] initialization
[Li et al. 2018a] initialization

Fig. 6. Deep inverse rendering from two input photographs starting from different initial starting points. A random starting point (2nd row) is more likely to generate non-plausible results compared to the reference (top row). Starting from the SVBRDFs generated by the single image estimation methods of Deschaintre et al. [2018] (3rd row) and Li et al. [2018a] (bottom row) produces similar plausible results.

holds, and thus this allows us to relax this condition and instead maximize an approximate proxy probability.

To bootstrap our optimization, we rely on prior learning-based single image SVBRDF estimation methods [Deschaintre et al. 2018; Li et al. 2018a]. These methods are trained to maximize $P(s|I_0)$, which is an acceptable approximation of $P(s|\{I_i\}_i)$ for many spatially-varying SVBRDFs. In practice, we found that our deep inverse rendering method is able to converge to a plausible solution from these initial SVBRDFs. Figure 6 shows a comparison for a wood material optimized from 5 input photographs with different initial starting points: (a) random starting point, (b) from [Deschaintre et al. 2018], and (c) from [Li et al. 2018a]. For comparison we also show visualizations lit from a novel lighting direction. As expected, our method fails to produce a plausible result from a random starting point, while achieving a plausible result when started from the single image SVBRDF estimates. Unless noted otherwise, we will use the method of Deschaintre et al. [2018] to initialize the optimization for the results shown in the remainder of the paper.

6 DETAIL REFINEMENT

A well known problem with fully convolutional neural encoder-decoder networks is that the information must pass through a so-called bottleneck. Unless the modeled space is sufficiently low dimensional, detail is lost. A common solution in convolutional networks is to include skip connections from the encoder to the decoder to inject the lost details back into the decoding stream. However, for our particular use of an auto-encoder where the latent space itself is the main goal, we cannot fall back on skip connections. As a consequence, our decoded SVBRDFs are less sharp. One potential solution would be to increase the size of the latent code. However, this would also act as a deregularizer on the latent code, leading to more objectionable artifacts in the optimized SVBRDFs. Instead, we propose to reintroduce details by performing an additional refinement post-processing step that directly adjusts the reflectance maps similar to classic inverse rendering as in Equation (1).

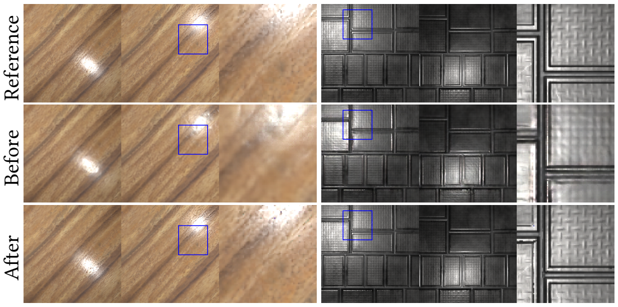

Fig. 7. A comparison of SVBRDFs recovered from 5 input photographs, before and after refinement, visualized from novel views. The visualizations without refinement lack sharpness and detail. Refinement reintroduces many of the missing details such as the speckle pattern in the specular highlight visible in the 2nd column (enlarged in 3rd column), and the tread plate pattern visible in the highlight in the 4th column (enlarged in the 6th column).

Typically, an unconstrained inverse rendering optimization is prone to introducing artifacts unless the starting point is already close to the solution. We argue that the solution from the deep inverse rendering framework represents such a starting point. Furthermore, since the solution from the deep inverse rendering framework is already a good solution, we only need a modest number of iterations to converge to a good solution. Figure 7 shows two examples of SVBRDFs before and after refinement using 5 input photographs. As can be seen, missing detail is reintroduced, while at the same time retaining most of the features from the deep inverse rendering SVBRDF.

7 RESULTS

7.1 Implementation Details

We implemented our framework in Tensorflow [Abadi et al. 2015], including a differentiable renderer constructed from built-in layers in Tensorflow. We use Adam [Kingma and Ba 2015] to train the auto-encoder using a learning rate of 10^{-4}, and β_1 set to 0.5; all other hyperparameters are kept to Tensorflow's defaults. We initialize the network for training with random values drawn from a zero mean normal distribution with a variance of 0.02. We train the auto-encoder in an end-to-end fashion for $100k$ iterations using mini-batches of 64 samples on the SVBRDF training dataset of Deschaintre et al. [2018]. Training takes about 16 hours on a workstation with dual NVidia GTX 1080Ti GPUs.

We also implemented the deep inverse rendering optimization in Tensorflow as it uses many of the same components as for the SVBRDF auto-encoder. We again use Adam as the optimization algorithm and set the learning rate to 10^{-3}, and $\beta_1 = 0.5$; all other hyperparameters are kept to the default. We ran Adam for $4k$ iterations, regardless of the resolution. The refinement is also implemented in

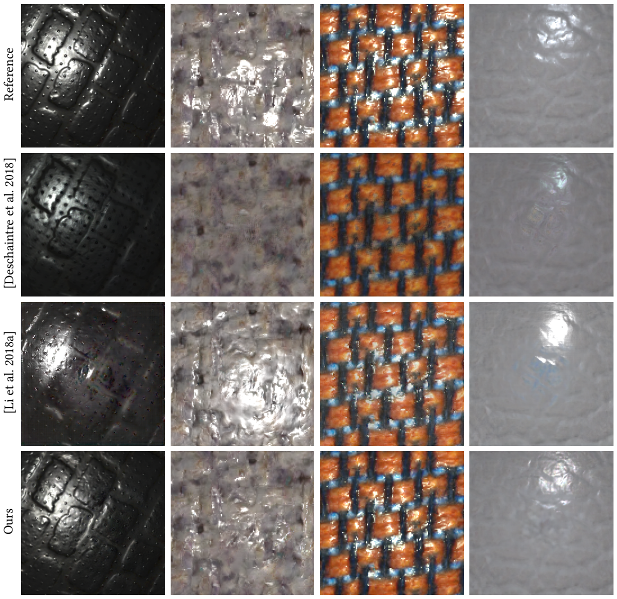

Fig. 8. Additional SVBRDFs estimated from a single input photograph and visualized under a novel lighting condition.

Tensorflow using Adam with the same hyperparameters. Because the refinement starts from the result of the deep inverse rendering, we can significantly reduce the number of iteration. For the results in this paper, we set the number of refinement iterations to 200.

Multi-resolution Optimization. The optimization computation cost increases proportionally with image resolution. To accelerate computations, especially for high resolution SVBRDFs, we implemented a multi-resolution optimization strategy. We start by downsampling the top-view to the native 256 × 256 resolution of the auto-encoder (which also is the native resolution of the single image SVBRDF networks used for initialization). We then run our deep inverse rendering algorithm as before, but only for $1k$ iterations and without refinement. The resulting property maps are then bilinearly upsampled to double resolution, and used as starting point for a double resolution deep inverse rendering optimization, again for $1k$ iterations. We continue this process until we reach the target resolution. For the last deep inverse rendering pass, we use $2k$ iterations to ensure good convergence. Finally, we run the refinement step for 200 iterations. This multi-resolution approach halves the computation time, and produces a comparable result to a full resolution optimization. We have used this method for all high resolution SVBRDFs in this paper.

7.2 Synthetic Acquisition Results

We first validate the quality and accuracy of our results on SVBRDFs from publicly available datasets that provide us with reference reflectance maps. We simulate the acquisition by randomly placing the camera/light at unit distance above the sample, and aimed at center of the exemplar.

Figure 9 compares deep inverse rendering results (with refinement) at 256 × 256 resolution for 1, 2, 5, and 20 input images, as well as against the results from prior single-image methods [Deschaintre et al. 2018; Li et al. 2018a]. Comparing the single image results for prior work and ours (rows 2−4) show that deep inverse rendering: recovers a more accurate match to the input photograph, provides more plausible results exhibiting less artifacts, and produces cleaner reflectance property maps. Figure 8 shows additional single image comparisons for a variety of materials, visualized under novel lighting conditions. On average, we found that deep inverse rendering further improves on the single image SVBRDF estimates. Increasing the number of input photographs (Figure 9, rows 5−7) shows progressively more accurate results compared to the reference. Note that 20 images is still a sparse sampling of all possible view and lighting directions.

Besides improving the reconstruction results, deep inverse rendering from more than one input photograph can help to correctly resolve ambiguous single image cases, i.e., two or more plausible SVBRDFs exist that can produce the same image. Figure 10 shows such a case. A single image reconstruction (2nd rows) might select the wrong 'most plausible' SVBRDF based on which initial SVBRDF was preferred by the single image SVBRDF network of Deschaintre et al. [2018]. However, as the number of input images increases, the ambiguity between the two cases begins to resolve. Figure 10 (3rd row) shows that with just 2 photographs we can already see convergence towards the correct solution. At 20 input images (last row) the ambiguity is mostly resolved.

We refer to the supplementary material for more results.

7.3 Real Acquisition Results

We also use our deep inverse rendering framework to recover high resolution (1024 × 1024) SVBRDFs from several real-world material samples. The input LDR photographs are captured using the backfacing camera of a mobile phone with the flash light turned on; we ensure that one photograph is captured from the top view. Each photograph is radiometrically linearized using a simple inverse gamma 2.2 correction. We also place a checkerboard around the sample to estimate the view (and lighting) direction.

Figure 11 shows, 3 captured materials, rerendered from two lighting directions and compared to reference photographs not part of the set used for deep inverse rendering. As can be seen, we achieve plausible results for the three materials and different number of input photographs. We refer to the supplementary material for reconstructions from a different number input photographs, as well as other materials.

8 DISCUSSION

Deep Inverse Rendering vs. Classic Inverse Rendering. In Section 5 we proposed to add a post-processing step to reintroduce the details that the auto-encoder cannot reproduce. This post-processing step is very similar to a classic inverse rendering optimization (albeit with much less iterations, and without additional regularization terms). We argue we do not need additional regularization terms because the solution from deep inverse rendering is generally accurate enough to support direct optimization without regularization. This raises the question whether direct optimization starting from

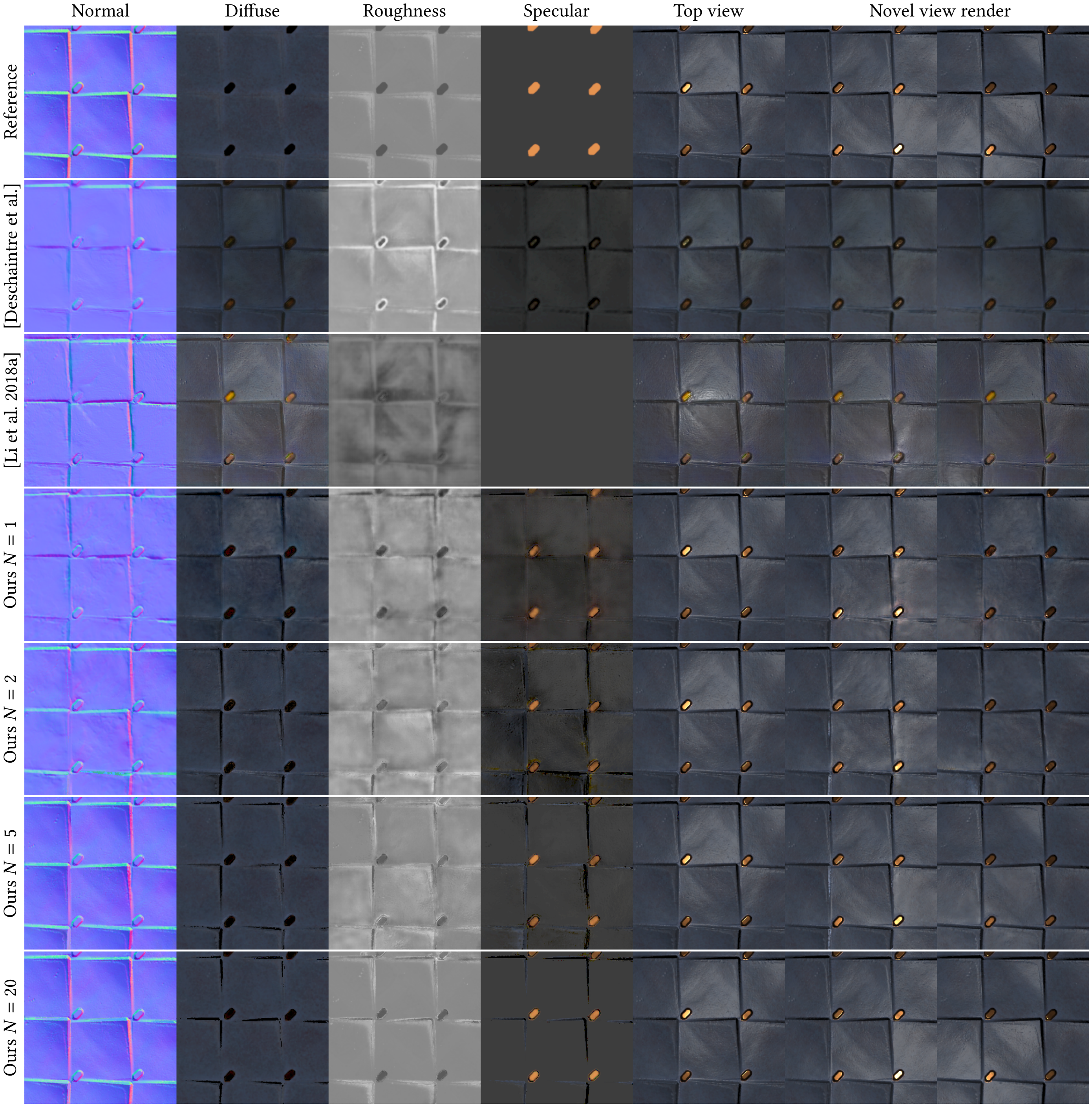

Fig. 9. Deep inverse rendering results for an increasing number of input photographs (rows 4 – 7) show an improvement in accuracy. Deep inverse rendering on a single input image also improves the quality of the results compared to the existing learning-based single image SVBRDF estimation methods [Deschaintre et al. 2018] (2nd row; also used as starting point for rows 4 – 7) and [Li et al. 2018a] (3rd row).

an SVBRDF provided by the single image estimation methods [Deschaintre et al. 2018; Li et al. 2018a] would also work? In other words, is the optimization in the latent space necessary and/or beneficial?

Figure 12 (bottom) compares 3 selected materials estimated from a varying number of input photographs using (a) deep inverse rendering, (b) deep inverse rendering + refinement, and (c) classic

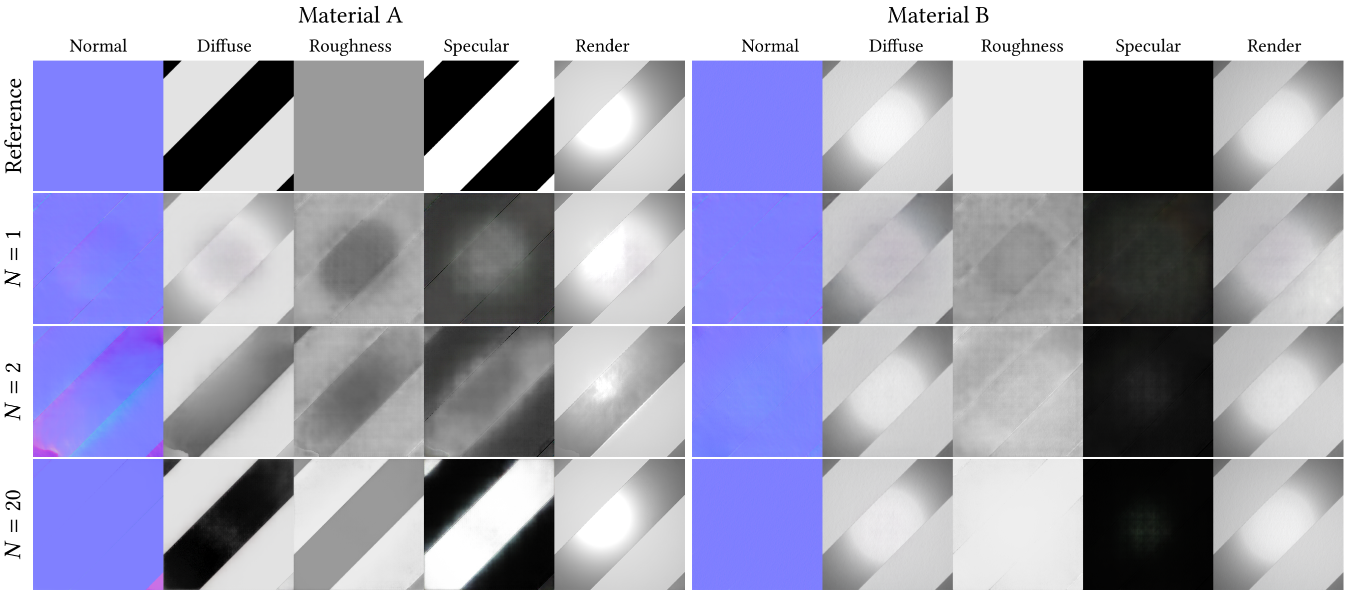

Fig. 10. Two ambiguous SVBRDFs that differ in their reflectance property maps, but produce similar images for the top view. Deep inverse rendering produces plausible results from one input photograph, but fails to resolve the inherent ambiguity. Increasing the number of photographs helps to resolve the ambiguities.

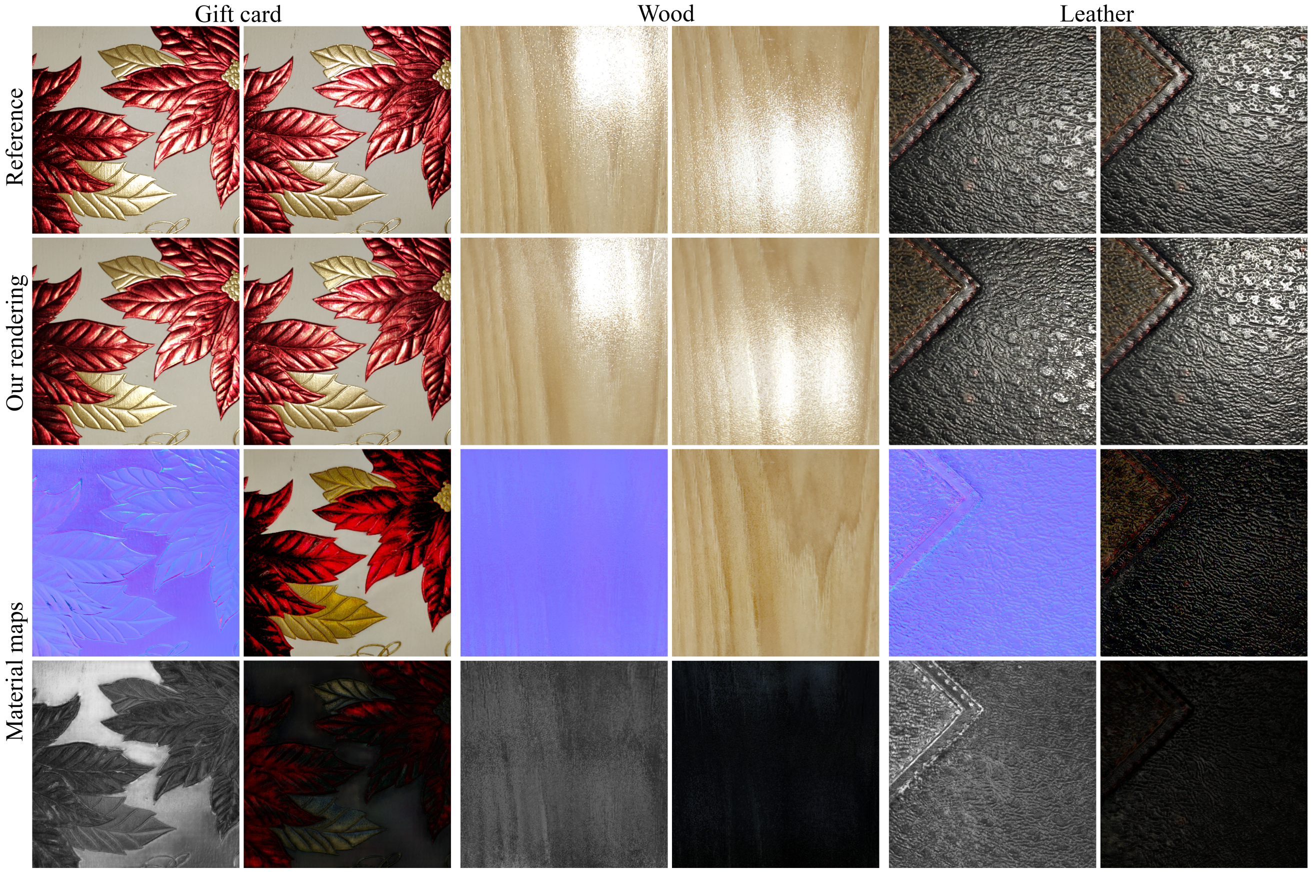

Fig. 11. Examples of three captured real-world material SVBRDFs reconstructed from 20 (Gift card), 10 (Wood), and 2 (Leather) input photographs. All the result maps (3rd and 4th row) are recovered at 1024 × 1024 resolution. We also compare rerenderings under novel lighting conditions (2nd row) with reference captured photographs (1st row).

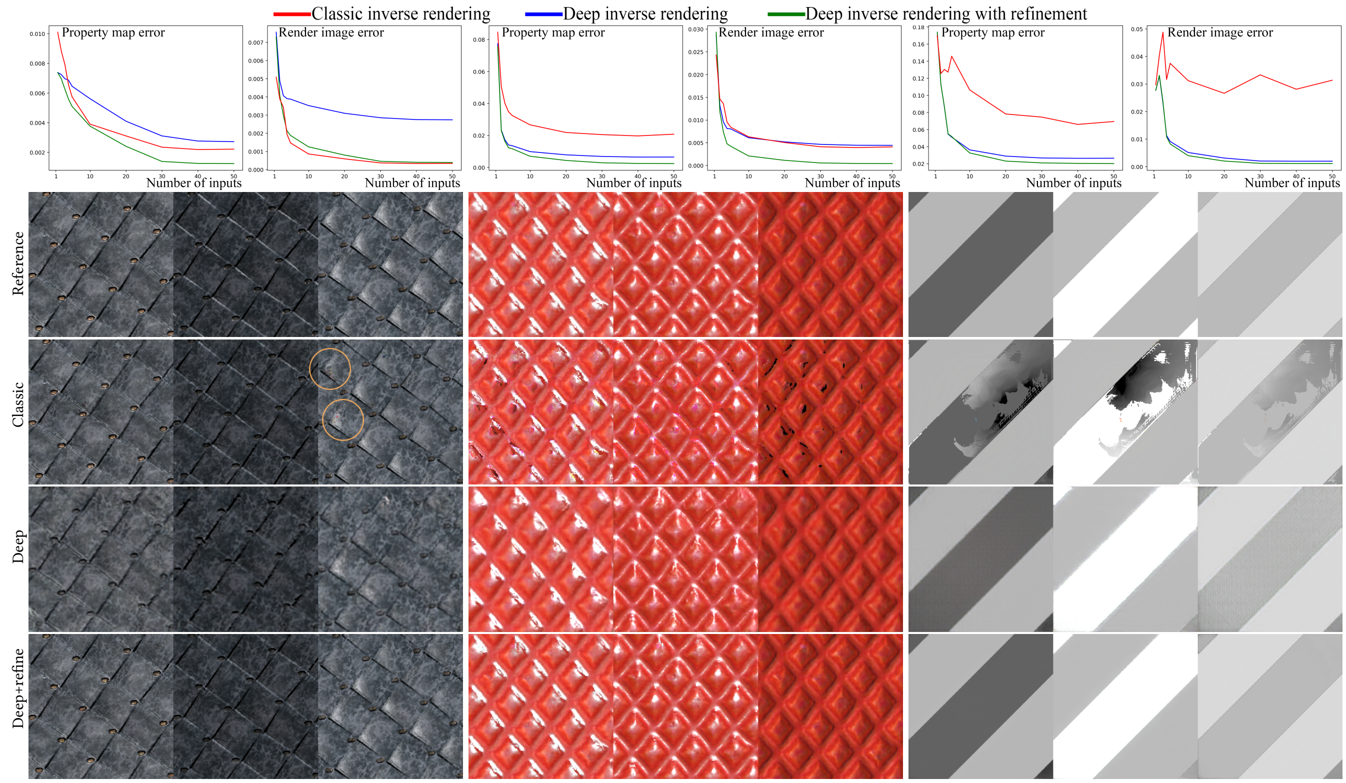

Fig. 12. Deep inverse rendering vs. classic inverse rendering illustrated on three selected materials. (top) Error plots of number of input photographs versus the L_2 error on the reflectance property maps (odd columns) and the L_2 error with respect to the rendered images (even columns) for classic inverse rendering (red), deep inverse rendering with (green) and without (blue) refinement. (bottom) Visualizations of the three selected materials under novel lighting directions for a different number of input images. We show the converged results (50 input photographs) for the middle and right materials. However, even at 50 input photographs, the classic inverse rendering results still contain artifacts. For the left material, we show the results for 10 images at which point the classic inverse rendering approach has a lower rendering error. Despite the lower render error for the left case, the visualizations of the classic inverse rendering method still exhibit subtle visual artifacts (circled), whereas the deep inverse rendering results do not.

inverse rendering all starting from the same starting point with an equal number of total iterations. We set the number of iterations large enough to ensure convergence for the classic inverse rendering method (i.e., less than 1% relative change in loss). Deep inverse rendering without refinement produces plausible results, albeit sometimes lacking in detail. An additional refinement step brings back these details. While, classic inverse rendering does sometimes succeed, in a significant number of cases, it fails with visually noticeable artifacts. Indicating that the starting point is too far from the target SVBRDF. As expected, when the number of images increases, and thus additional constraints are introduced, classic inverse rendering produces less artifacts. However, we found that deep inverse rendering with refinement produces results without artifacts earlier (i.e., with less input photographs), as illustrated in the error graphs in Figure 12 (top) that plots the reconstruction error for a varying number of input images. The graphs plot the L_2 error on the reflectance property maps (odd columns) and the L_2 error on the rendered images (even columns). Due to the non-linear mapping from reflectance properties to rendered images, these two types of errors are not always consistent. For example, a minor error in the normal map (i.e., low reflectance map error) can lead to large visual differences for highly specular materials (i.e., large render error). Conversely, a large specular roughness combined with a low specular albedo can lead to little visual difference. Classic inverse rendering (red curve) generally produces *noisy* reflectance property maps, and thus a larger map error. Even when the rendering error is low, it often still includes visually noticeable artifacts. Deep inverse rendering (blue curve) produces SVBRDFs with less artifacts, but often lacking in detail. This impacts the error on the reflectance maps and renderings (depending on the sharpness of the features in the underlying SVBRDF). In the majority of the cases, deep inverse rendering without refinement outperforms classic inverse rendering. Deep inverse rendering combined with refinement (green curve) performs consistently best.

We further quantitatively analyze the differences in accuracy between deep inverse rendering (with and without refinement) and classic inverse rendering on a large synthetic dataset. To avoid bias towards the characteristics of the training dataset, for both the initialization network as well as our auto-encoder, we compose the test set of 42 SVBRDFs of which 20 SVBRDFs are from the test set

Table 1. Quantitative comparison for different number of input images of L_2 reflectance map error and L_2 rendering error (over 100 random view and lighting directions) for a large synthetic test set of 42 SVBRDFs. The best (i.e., lowest) error for each component and number of input images is marked in bold.

	Classic inverse rendering						Deep inverse rendering						Deep inverse rendering with refinement					
N	Diffuse	Specular	Roughness	Normal	Map Average	Render Error	Diffuse	Specular	Roughness	Normal	Map Average	Render Error	Diffuse	Specular	Roughness	Normal	Map Average	Render Error
1	0.016030	**0.02109**	0.08772	**0.003790**	0.03215	0.007594	**0.014400**	0.02123	0.07209	0.004214	0.02798	0.007443	0.014440	0.02121	**0.07145**	0.004235	**0.02783**	**0.007394**
2	0.009133	0.01818	0.07673	0.003293	0.02684	0.006141	0.006029	0.01725	0.06269	0.002284	0.02206	0.004717	**0.005919**	**0.01722**	**0.06235**	**0.002079**	**0.02189**	**0.004369**
5	0.006182	0.01485	0.06686	0.001340	0.02231	0.002163	0.003349	0.009078	**0.05515**	0.001042	0.01716	0.002503	**0.002854**	**0.00817**	0.05548	**0.000489**	**0.01675**	**0.001437**
20	0.003658	0.00724	0.05278	0.001198	0.01622	0.001092	0.002098	0.006228	0.04249	0.000715	0.01288	0.001637	**0.000850**	**0.00447**	**0.04130**	**0.000273**	**0.01172**	**0.000413**

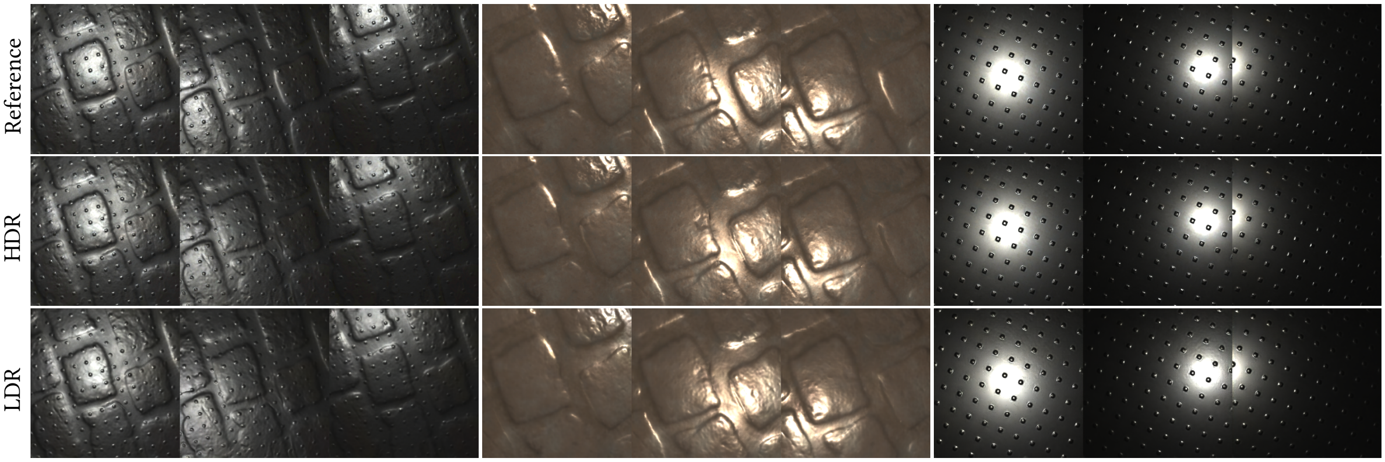

Fig. 13. A comparison of SVBRDF estimations for three selected materials from 2 (left column), 5 (middle column), and 20 (right column) input HDR (2nd row) and LDR (last row) photographs. The LDR results are comparable to the HDR results, with some subtle details missing due to the quantization of the input photographs.

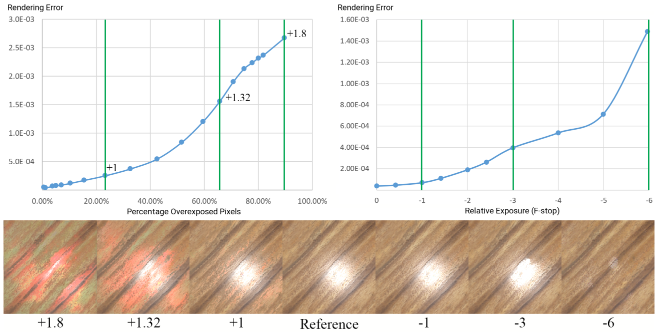

Fig. 14. Over and undersaturation in LDR input photographs results in a loss of information that can affect the reconstruction of the specular component. (top-left) Rendering error increases when the ratio of oversaturated pixel increases, resulting in color artifacts in the specular highlights. (top-right) Reducing the exposure increases the number of pixels affected by quantization artifacts, resulting is a dissolution of the specular component. (bottom) Visualization of the reconstructed SVBRDF under a novel lighting condition for selected exposures (expressed in F-stops and marked by a green line in the graphs).

provided by Deschaintre et al. [2018], 15 SVBRDFs are from Aittala et al. [2015], 4 are from the Free PBR Materials website [2], and 3 are manually created; we refer to the supplemental material for a detailed listing. We crop each SVBRDF to a 256 × 256 resolution, and generate sets of 1, 2, 5, and 20 synthetic input images for randomly selected view/lighting directions. Table 1 summarizes the average L_2 errors on each of the reflectance property maps and the rendering error over 100 novel view and lighting directions. While in a few cases, classic rendering and/or deep inverse rendering without refinement produces a smaller error on an isolated property map, in general, deep inverse rendering with refinement leads to a lower rendering error. This is consistent with the results in Figure 12 that showed higher visual fidelity for deep inverse rendering with refinement.

LDR vs. HDR Input Photographs. At its core, our method is an inverse rendering method. Therefore, it is straightforward to adapt our optimization to reconstruct the SVBRDF from either low dynamic range (LDR; 8-bit linear images) or high dynamic range (HDR) photographs; we only need to clamp pixel values to [0, 1] and add a quantization step after rendering for LDR optimization. Figure 13 compares for 3 selected materials the optimized SVBRDF results from LDR and HDR photographs. In general, the results are qualitatively very similar. We observe that the LDR images sometimes suffer from additional artifacts, especially when small pixel values get quantized to 0. In general, the recovered LDR SVBRDFs are remarkably accurate even when some of the peaks of the highlights are clamped. However, this raises the question to how much oversaturation our method can bear. Figure 14 (top-left) plots the rendering error with respect to the ratio of well exposed versus oversaturated pixels for $N = 20$ input images. From this we can see that high

[2] https://freepbr.com

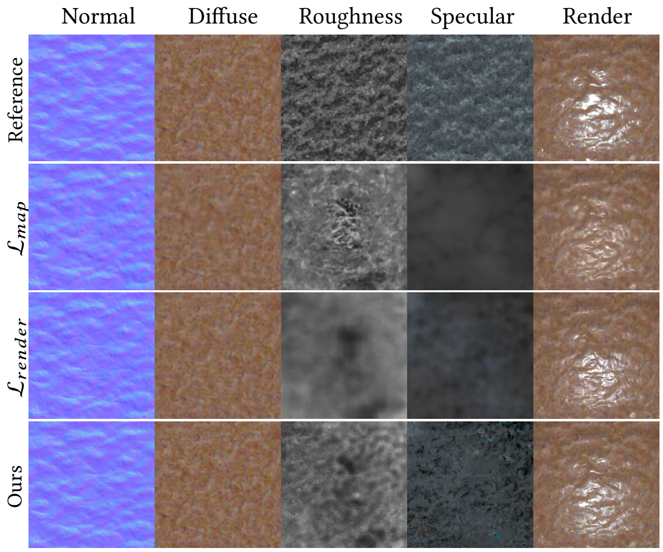

Fig. 15. Impact of the auto-encoder training loss on deep inverse rendering without refinement demonstrated for two input photographs. While the auto-encoder trained with only the reflectance map loss $\mathcal{L}_{map}$ produces sharper property maps, the corresponding visualizations do not reflect the expected appearance. Training using only the render loss $\mathcal{L}_{render}$ produces more blurry property maps, specially for the roughness and specular albedo maps. Our combined loss function strikes a balance between map detail and render accuracy.

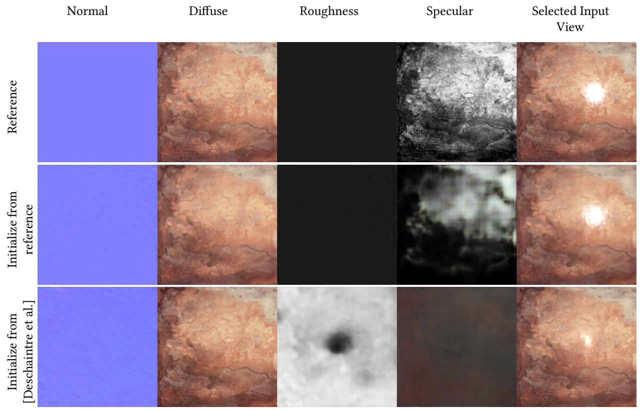
Normal
Diffuse
Roughness
Specular
Selected Input View
Reference
Initialize from reference
Initialize from [Deschaintre et al.]

Fig. 16. An example of a failure case caused by a suboptimal starting point. Starting from the reference SVBRDF, without refinement, produces an accurate reconstruction, albeit with some detail missing in the specular albedo map (2nd row). This indicates that the SVBRDF lies in the latent embedding. However, when starting from a suboptimal SVBRDF (from [Deschaintre et al. 2018]), deep inverse rendering fails to correct the specular roughness and albedo maps (last row).

quality results can be estimated when 20% of the pixels are oversaturated in the input. Increasing the ratio of oversaturated pixels further results in visual artifacts in the specular highlights (Figure 14 (bottom)). Conversely, we also explore the impact of quantization on underexposed inputs, i.e., pixels are mapped to a low or zero pixel value (Figure 14 (top-right)). In general, we observe less artifacts in the recovered SVBRDFs, except for severely underexposed images (-6 F-stops) where the specular component disappears.

Training Loss Function. Prior work has used the loss on the reflectance maps [Li et al. 2017], the loss on renderings of the materials [Deschaintre et al. 2018], or a combination of both [Li et al. 2018a]. However, the goal of these prior works is different than ours as they aim for training an inference network as opposed to learning a space suitable for inverse rendering. We therefore evaluate all three in the context of deep inverse rendering. Figure 15 shows the results of optimizing an SVBRDF using an auto-encoder trained with three different loss functions ($\mathcal{L}_{map}$, $\mathcal{L}_{render}$, and our training loss $\mathcal{L}_{map} + \frac{1}{9}\mathcal{L}_{render}$) from two input photographs. From this we can see that using only $\mathcal{L}_{map}$ tends to produce sharper reflectance maps. However, this detail does not always result in plausible visualizations. Using only $\mathcal{L}_{render}$ produces overly blurred reflectance maps, especially the specular roughness and albedo maps. Our training loss function strikes a balance between regularizing and retaining detail.

Impact of Initialization Accuracy. Figure 16 shows a failure case of our deep inverse rendering framework. We argue that the failure is caused by an initialization that is insufficiently accurate/plausible. We can easily show that the SVBRDF of this failure case can be accurately represented in the latent space by encoding the reference SVBRDF, and further optimizing it using deep inverse rendering (without refinement) to address shortcomings in the encoder. While the reconstruction is less detailed, a sufficiently accurate representation can be found in the latent space. This indicates that deep inverse rendering does not find the correct solution and ends up in a local minimum. Our method is at its core a nonlinear optimization due to the nonlinear mapping from latent code to rendered images. Like any nonlinear optimization, a suboptimal starting point will lead to a local minimum. In general, we observe that deep inverse rendering has trouble correcting the initial starting point when the roughness values are too large combined with an underestimation of the specular albedo. We posit that this creates a strong local minimum because small changes in roughness affect a large area in the renderings (producing a global error), while only a yielding a small change in each pixel's value. Hence, if there is only a localized error, e.g., a missing highlight, then it will be difficult to balance the global error introduced on all pixels versus the small local improvement on the highlight; increasing the number of photographs does not resolve this problem.

Currently, we rely on a single image estimation method [Deschaintre et al. 2018; Li et al. 2018a] to provide a starting point. If the input image used for initialization does not exhibit a visual effect from a reflectance component, or if it has ambiguous reflectance features, then such an initialization can be suboptimal. However, one of the strengths of our method is that it is not married to a particular initialization method, and future advances (e.g., generalizations of Kim et al. 's [2017] method to SVBRDFs) can be easily used for bootstrapping our method. Even in the case of future advances in deep multiview SVBRDF estimation solutions, our *deep inverse rendering* method still offers an advantage in that it optimizes to the input

Table 2. Quantitative impact of calibration errors on the light position (expressed in degrees with respect to the direction towards the center of the sample) and the impact of environment lighting (expressed by relative power with respect to the point light power).

	Diffuse	Specular	Roughness	Normal	Map Average	Render Err.
1°	0.000901	0.004756	0.04143	0.000281	0.01184	0.000496
5°	0.001578	0.005981	0.04492	0.000365	0.01321	0.001144
10°	0.002427	0.007834	0.05208	0.000642	0.01575	0.002654
0°/ 0%	0.000850	0.004470	0.04130	0.000273	0.01172	0.000413
1%	0.000809	0.005403	0.04073	0.000280	0.01181	0.000481
5%	0.001663	0.005546	0.04033	0.000306	0.01196	0.000929
10%	0.003577	0.008345	0.04162	0.000374	0.01348	0.002270

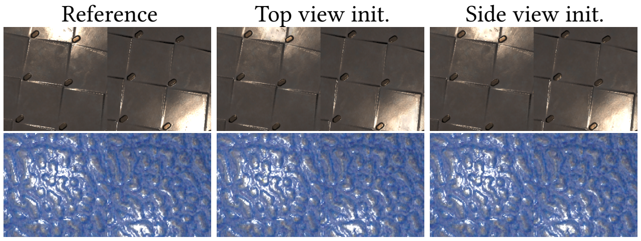

Fig. 17. Comparison of visualizations of SVBRDFs estimated from 5 (top) and 20 (bottom) input photographs bootstrapped from the top view or side view (45°). Due to the robustness of the single image SVBRDF estimation method of Deschaintre et al. [2018], we can relax the condition that the top view needs to be captured.

instead of to the average loss over a training dataset, as well as its ability to optimize high resolution solutions from low resolution initializations.

Top View Constraint Relaxation. If the method used to estimate the initial starting point SVBRDF is robust to deviations from the top view (e.g., such as [Deschaintre et al. 2018]), then we can relax the condition that at least one photograph must be captured from the top view. Because our framework is based on inverse rendering, it does not require retraining to accommodate this change. Figure 17 shows deep inverse rendering results initialized from a side view at 45 degrees. Note that we removed the top view from the set of input photographs. The SVBRDFs obtained from both 5 and 20 photographs are virtually identical to the top view initialization.

Lighting Robustness. Our method assumes that the SVBRDF sample is lit only from a point light colocated with the camera, and that its position is known. We perform two experiments to gain better insight on the impact on the accuracy if these preconditions are not met.

In a first experiment, we add random perturbations to the light source position in each of the $N = 20$ input images. We limit the degree of perturbations such that the direction of the light, measured relative to the center of the sample, falls within a predefined cone. As demonstrated in Figure 18 (rows 1-4), an error in the light position results in errors in the surface normal, and consequently, also in the specular component. Table 2 summarizes the average errors over the test set of 42 SVBRDFs for 1, 5, and 10 degrees of error. Both Figure 18 and Table 2 indicate that our method is able to produce good results for errors up to 10 degrees.

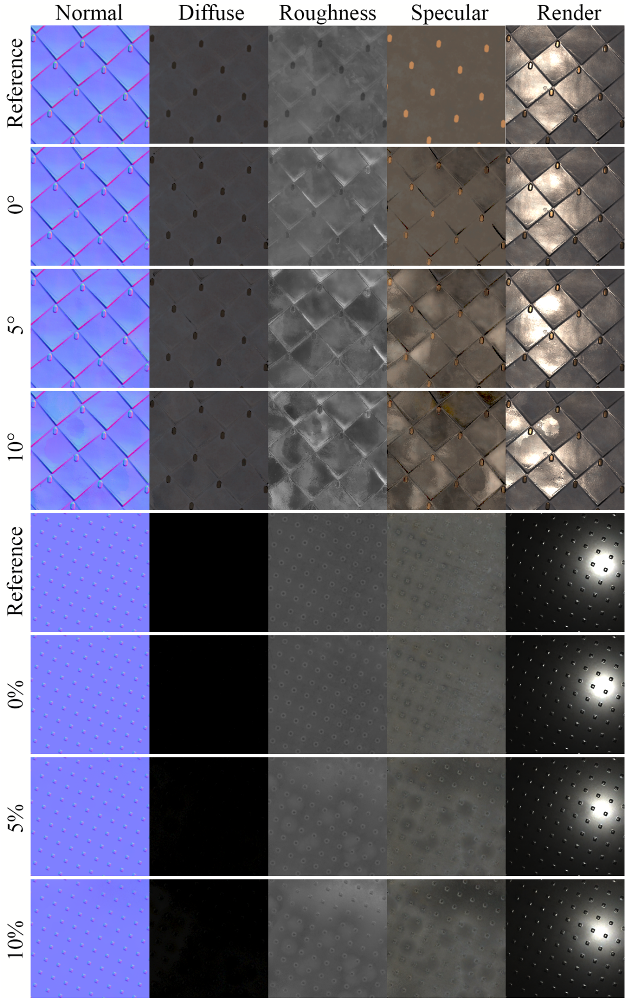

Fig. 18. Robustness of our method to lighting position error and to uncontrolled environment light contamination. The top four rows show the reconstruction results and visualizations under novel lighting based on accurate light position (2nd row) and with a random position error within a 5° (3rd row) and a 10° (4th row) range. The bottom four rows show the results with no environment lighting (6th row), with environment lighting at 5% of the point light intensity (7th row), and with 10% of the point light intensity (8th row). Reference SVBRDFs and visualizations are shown in the 1st and 5th row.

As a second experiment, we validate the robustness of our method to deviations from the ideal point lighting by including environment lighting (i.e., *Uffizi Gallery*) scaled to control the relative brightness of the point light versus the environment lighting. During deep inverse rendering, we ignore the environment lighting. Figure 18 (rows 5-8), show that uncontrolled environment lighting affects the specular component by baking in the specular highlight from the environment lighting into the specular albedo and roughness maps. Table 2 quantifies the errors over the test dataset of 42 SVBRDFs, showing that our method is robust to moderate degrees of environment lighting.

9 CONCLUSIONS

In this paper we presented a novel unified framework for high resolution SVBRDF estimation using inverse rendering from an arbitrary number of photographs. The precision of the estimated SVBRDFs automatically adapts to the number of input photographs, ranging from plausible estimates for underconstrained acquisitions (e.g., a single photograph) to accurate reconstructions for fully constrained conditions. Our framework does not rely on fragile handcrafted heuristics or regularization terms, but instead directly optimizes learned features. We achieve this by optimizing in a latent embedding of the space of SVBRDFs learned by an auto-encoder. We propose a number of enhancements to regularize the learned latent space to facilitate optimization. We demonstrated that our framework is suitable for estimating high resolution SVBRDFs from an arbitrary number of input photographs. Furthermore, we show that our method can improve the quality of existing deep learning based single image SVBRDF estimation methods.

For future work we would like to improve on the initialization of the deep inverse rendering optimization. Currently we rely on existing methods that are designed for producing visually good results and not for initializing a deep inverse rendering optimization. Joint optimization of an initialization network as well as the auto-encoder for optimization is an interesting avenue for future research.

ACKNOWLEDGMENTS

We would like to thank the reviewers for their constructive feedback. We also thank Baining Guo for discussions and suggestions. Pieter Peers was partially supported by NSF grant IIS-1350323 and gifts from Google, Activision, and Nvidia. Duan Gao and Kun Xu are supported by the National Natural Science Foundation of China (Project Numbers: 61822204, 61521002).

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/

Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2016. Reflectance Modeling by Neural Texture Synthesis. *ACM Trans. Graph.* 35, 4, Article 65 (July 2016).

Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2015. Two-shot SVBRDF Capture for Stationary Materials. *ACM Trans. Graph.* 34, 4, Article 110 (July 2015).

Dan A. Calian, Jean-François Lalonde, Paulo Gotardo, Tomas Simon, Iain Matthews, and Kenny Mitchell. 2018. From Faces to Outdoor Light Probes. *Computer Graphics Forum* 37, 2 (2018), 51–61.

Inchang Choi, Daniel S. Jeon, Giljoo Nam, Diego Gutierrez, and Min H. Kim. 2017. High-Quality Hyperspectral Reconstruction Using a Spectral Prior. *ACM Trans. Graph.* 36, 6, Article 218 (2017).

Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. 2018. Single-Image SVBRDF Capture with a Rendering-Aware Deep Network. *ACM Trans. Graph.* 37, 128 (aug 2018).

Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. 2014. Appearance-from-motion: Recovering Spatially Varying Surface Reflectance Under Unknown Lighting. *ACM Trans. Graph.* 33, 6, Article 193 (2014).

Julie Dorsey, Holly Rushmeier, and Franois Sillion. 2008. *Digital Modeling of Material Appearance.* Morgan Kaufmann Publishers Inc.

Leon Gatys, Alexander S Ecker, and Matthias Bethge. 2015. Texture synthesis using convolutional neural networks. In *NIPS*. 262–270.

Geoffrey Hinton and Ruslan Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504 – 507.

Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, and Aswin Sankaranarayanan. 2017. Reflectance Capture using Univariate Sampling of BRDFs. In *ICCV*.

Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*. 448–456.

Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. 2018. Efficient Reflectance Capture Using an Autoencoder. *ACM Trans. Graph.* 37, 4, Article 127 (July 2018).

Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias Nießner, and Jan Kautz. 2017. A Lightweight Approach for On-the-Fly Reflectance Estimation. In *ICCV*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2017. Modeling Surface Appearance from a Single Photograph Using Self-augmented Convolutional Neural Networks. *ACM Trans. Graph.* 36, 4, Article 45 (July 2017), 11 pages.

Zhengqin Li, Kalyan Sunkavalli, and Manmohan Krishna Chandraker. 2018a. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. *ECCV*.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2018b. Learning to Reconstruct Shape and Spatially-varying Reflectance from a Single Image. *ACM Trans. Graph.* 37, 6 (2018), 126.

Gianpaolo Palma, Marco Callieri, Matteo Dellepiane, and Roberto Scopigno. 2012. A Statistical Method for SVBRDF Approximation from Video Sequences in General Lighting Conditions. *Comput. Graph. Forum* 31, 4 (2012), 1491–1500.

Jérémy Riviere, Pieter Peers, and Abhijeet Ghosh. 2016. Mobile Surface Reflectometry. *Comput. Graph. Forum* 35, 1 (2016), 191–202.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing High-Dimensional Data Using t-SNE. (2008), 2579–2605.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. In *Rendering Techniques*. 195–206.

Michael Weinmann and Richard Klein. 2015. Advances in Geometry and Reflectance Acquisition. In *ACM SIGGRAPH Asia, Course Notes*.

Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. 2016. Recovering Shape and Spatially-Varying Surface Reflectance under Unknown Illumination. *ACM Trans. Graph.* 35, 6 (December 2016).

Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. 2016. Minimal BRDF Sampling for Two-shot Near-field Reflectance Acquisition. *ACM Trans. Graph.* 35, 6, Article 188 (Nov. 2016).

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. 2018. Deep image-based relighting from optimal sparse samples. *ACM Trans. Graph.* 37, 4 (2018), 126.

Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2018. Single Photograph Surface Appearance Modeling with Self-Augmented CNNs and Inexact Supervision. *Comput. Graph. Forum* 37, 7 (Oct 2018).

Zhiming Zhou, Guojun Chen, Yue Dong, David Wipf, Yong Yu, John Snyder, and Xin Tong. 2016. Sparse-as-Possible SVBRDF Acquisition. *ACM Trans. Graph.* 35 (November 2016).