DOI: 10.1111/cgf.13920
EUROGRAPHICS 2020 / U. Assarsson and D. Panozzo
(Guest Editors)
Volume 39 (2020), Number 2

DeepBRDF: A Deep Representation for Manipulating Measured BRDF

Bingyang Hu[1], Jie Guo[1†], Yanjun Chen[1], Mengtian Li[1] and Yanwen Guo[1,2]

[1]State Key Lab for Novel Software Technology, Nanjing University, China
[2] iQIYI Intelligence, China

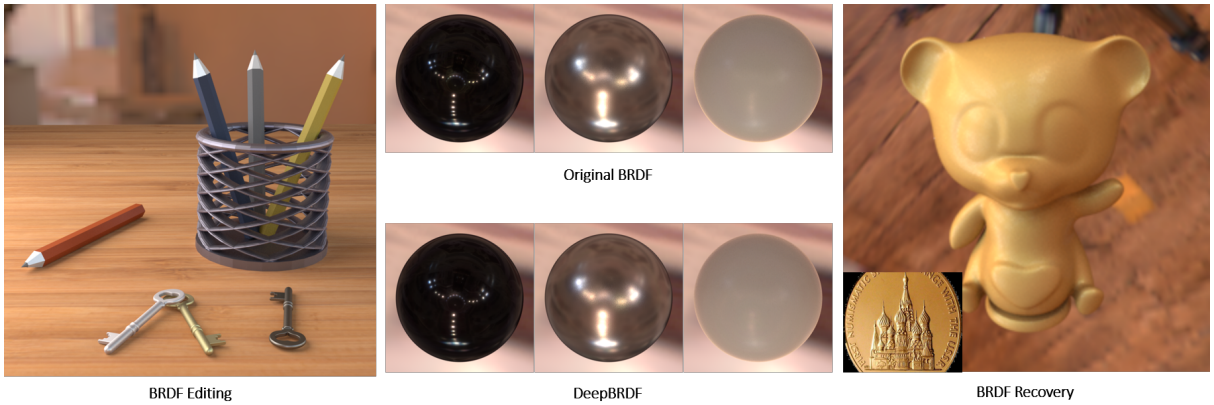

Figure 1: *The proposed DeepBRDF encodes measured BRDF data on a nonlinear low-dimensional manifold where the reconstructed BRDFs match the original ones (middle). It allows high quality BRDF editing (left) and accurate BRDF recovery from a single image (right).*

Abstract

Effective compression of densely sampled BRDF measurements is critical for many graphical or vision applications. In this paper, we present DeepBRDF, a deep-learning-based representation that can significantly reduce the dimensionality of measured BRDFs while enjoying high quality of recovery. We consider each measured BRDF as a sequence of image slices and design a deep autoencoder with a masked L_2 loss to discover a nonlinear low-dimensional latent space of the high-dimensional input data. Thorough experiments verify that the proposed method clearly outperforms PCA-based strategies in BRDF data compression and is more robust. We demonstrate the effectiveness of DeepBRDF with two applications. For BRDF editing, we can easily create a new BRDF by navigating on the low-dimensional manifold of DeepBRDF, guaranteeing smooth transitions and high physical plausibility. For BRDF recovery, we design another deep neural network to automatically generate the full BRDF data from a single input image. Aided by our DeepBRDF learned from real-world materials, a wide range of reflectance behaviors can be recovered with high accuracy.

CCS Concepts

• ***Computing methodologies*** → ***Reflectance modeling;*** *Neural networks;*

1. Introduction

With the development of novel gonioreflectometers, an increasing amount of data from real-world materials has been created [MPBM03, FV14, DJ18]. Although more accurate by definition, measured or data-driven BRDF models suffer from a high memory footprint and significant computational cost stemming from the storage and processing of massive BRDF data. This makes either editing or manipulating measured BRDFs quite difficult [DRS08, GGG*16].

Generally, the raw data of measured BRDFs can have an arbitrarily high dimensionality as many incident and viewing di-

† Corresponding author: jieguo@nju.edu.cn

rections are organized to capture and tabulate a BRDF. However, the actual subspace of real-world BRDFs is of substantially low dimensionality. Therefore, lots of previous methods try to reduce the dimensionality of the measured BRDF to facilitate its usage. Existing approaches conquer this challenge via either projecting the raw data onto a subset of linear principal components [MPBM03, NDM06, NJR15, XNY*16, SGM*16], or applying a nonlinear dimensionality reducer [MPBM03, WAKB09]. The former is efficient but requires a large number of components to reproduce the original BRDF with high fidelity. The latter is considered more robust at the cost of very complex BRDF interpolation and extrapolation. Another widely used strategy to lower the complexity of measured BRDFs is via fitting low-dimensional analytical models controlled by a set of parameters to the raw data [NDM05,LKYU12,BSH12,RBMS17]. Unfortunately, the fitting process, usually relying on nonlinear optimization, is often time-consuming and numerically unstable.

Inspired by the success of Convolutional Neural Networks (CNNs) in many fields, we propose *DeepBRDF*, a deep-learning-based representation to explore the low-dimensional subspace of real-world BRDFs in this paper. Specifically, we design a deep autoencoder [HS06,Ben09] to learn the latent space of measured data in the MERL BRDF dataset [MPBM03]. The autoencoder takes the full data of a measured BRDF as the input and try to reconstruct it after it passes through a "bottleneck" layer. By constraining the bottleneck to have a low dimension, the goal of dimensionality reduction for the measured BRDFs is achieved. Unlike PCA (Principal Component Analysis), the autoencoder is able to explore a nonlinear low-dimensional manifold when a nonlinearity such as a ReLU activation is used in the encoder. Our results demonstrate that the proposed DeepBRDF clearly outperforms PCA-based strategies in BRDF data compression and is quite efficient by taking advantage of recent advancements in deep learning.

We further show the benefits of the proposed DeepBRDF with the following two applications.

BRDF Editing. BRDF editing is straightforward for analytical BRDF models but is difficult for measured data. By building a mapping between some perceptual attributes, either low-level [GGPL18] or high-level [SGM*16], and the underlying DeepBRDF, we can easily perform BRDF editing for measured BRDFs. Compared against the PCA-based techniques, our method always reproduces physically plausible BRDFs and supports smooth interpolation/extrapolation.

BRDF Recovery. Recent studies have proven the capability of CNNs to estimate the parameters of analytical BRDFs from input natural images [KGT*17, LDPT17, MMZ*18, LXR*18, YLD*18, KCW*18, GLD*19]. However, the accuracy of the estimation is highly restricted by the scope of selected analytical BRDF models. With our DeepBRDF learned from real-world materials, a wider range of reflectance behaviors can be recovered. Based on the deep representation, we design a new CNN to automatically recover the full BRDF data of an arbitrary material from a single input image. Several experiments on both synthetic and real-world images demonstrate the effectiveness and robustness of the proposed method.

2. Related Work

Data-Driven BRDF Models. Data-driven BRDF models leverage measured data from real-world materials to encode appearances. Although significant progress has been made [NJR15, XNY*16], accurate BRDF measurements are still challenging due to mechanical and computational complexity [NDM05, GGG*16]. Currently, the most widely used BRDF datasets include MERL for isotropic materials [MPBM03] and UTIA for anisotropic ones [FV14]. The MERL dataset [MPBM03] contains 100 real-world materials covering a wide range of appearances. Each material consists of measurements from a dense set of directions. Since its publication, this dataset has led to better understanding of realistic materials [Bur12, GGPL18, LMS*19], and inspired researchers to derive more precise analytic models to match the measurements [LKYU12, BSH12, RBMS17]. Although the UTIA dataset [FV14] has more BRDFs and covers anisotropy, it is limited to low-frequency appearances. Most recently, a new dataset covering all-frequency spectral BRDF measurements is released [DJ18].

Low-Dimensional Representations. To ease the usage of densely measured high-dimensional BRDFs, dimensionality reduction, e.g., PCA [MPBM03,NDM06,NJR15,XNY*16,SGM*16], is usually required in many applications. Matusik et al. [MPBM03] proposed to perform BRDF dimensionality reduction by either linear analysis or nonlinear dimensionality reducers, resulting in a 45D or 15D manifold. Nielsen et al. [NJR15] improved traditional PCA by employing a log-relative mapping on raw BRDF values. To ensure nonnegativity, nonnegative matrix factorization [LRR04, LBAD*06] and homomorphic factorization [MAA01] have been applied to decompose high-dimensional BRDFs. Tensor decompositions have also proved successful in reducing the dimensionality of measured BRDFs [RK10,BÖK11]. Soler et al. [SSN18] explored a low-dimensional nonlinear BRDF manifold via a Gaussian process latent variable model, supporting smooth transitions across different BRDFs. Zsolnai-Fehér et al. [ZFWW18] proposed learning algorithms to recommend new materials and create a low-dimensional latent space for fine-tune, enabling rapid mass-scale material synthesis for any principled shader. In contrast, we learn a latent space for measured materials.

BRDF Editing. BRDF editing is straightforward once a measured BRDF is fitted to analytical models [CPK06, SZC*07, NKLN10, SPN*16]. Unfortunately, the range of reflectance behaviors is quite limited by the underlying analytical model. Therefore, it is more attractive to edit the measured BRDFs directly by either interpolating between materials in a dataset [MPBM03, WAKB09, SSN18] or alternating factorized low-dimensional curves [LBAD*06, BAOR06]. To allow intuitive editing, Serrano et al. [SGM*16] established the relationship between a handful of high-level perceptual attributes and the PCA coefficients of measured BRDFs. The approach proposed by Tsirikoglou et al. [TKL*16] supports intuitive and efficient editing by changing the differences between the edited analytical BRDF and original analytical BRDF fitted to the data. Several recent studies [GGPL18,SJR18] show that a proper diffuse-specular separation is beneficial for measured BRDF editing. We demonstrate that the proposed DeepBRDF facilitates BRDF editing. Compared with PCA-

based editing, e.g., [SGM*16], our method is more robust and physically plausible.

BRDF Recovery. The problem of BRDF recovery aims to infer reflectance behaviors from 2D images. Due to the inherent ambiguity between the intensity of lighting and surface reflectance, many optimization-based methods attempt to solve this problem by imposing simplistic assumptions on either geometries [ZREB06] or illumination [RZ10, BM15]. Undoubtedly, these assumptions or priors significantly limit the applicability of the proposed methods. Currently, the most successful solutions rely on deep learning. Several deep neural networks are only designed to infer diffuse material properties from a single image [TSH12, ZKE15, NMXY15, KWKT15]. Rematas et al. [RRF*16] predicted the combination of BRDF and illumination from a single image, handling both diffuse and specular materials. Liu et al. [LCY*17] presented a differentiable rendering layer to handle both diffuse and specular materials. Meka et al. [MMZ*18] suggested using a highly efficient perceptual rendering loss in training, leading to real-time estimation of an object's material. There are also some deep neural networks tailored to estimate spatially-varying BRDFs [LDPT17, LXR*18, YLD*18, KCW*18, GLD*19]. Rather than predicting parameters of analytical BRDF models, we propose to recover the densely sampled BRDF data directly from an input image based on DeepBRDF.

3. DeepBRDF

This section thoroughly discusses DeepBRDF, a low-dimensional representation of measured BRDFs.

3.1. Data Preprocessing

We currently work on the MERL BRDF dataset [MPBM03]. This dataset covers 100 isotropic materials of varying reflectances, ranging from soft diffuse materials like rubber to hard specular materials like chrome. The raw data are stored in 3D tables using Rusinkiewicz half-difference angle coordinates $(\theta_h, \theta_d, \phi_d)$ [Rus01]. The resolution for each color channel is $90 \times 90 \times 180$, leading to 1,458,000 measurements.

Note that the raw BRDF data usually contain high dynamic range (HDR) information. This means the magnitudes of input values can be arbitrarily high, which causes problems for most CNNs. To tackle this issue, we use a data preprocessing method similar to Log-Relative Mapping [NJR15]. Specifically, we use the following formula to compress the HDR data:

$$\rho = \ln\left(\frac{\rho+\varepsilon}{\rho_{\text{ref}}+\varepsilon}+1\right) \tag{1}$$

where ρ is the BRDF, $\varepsilon = 0.001$ is a small constant avoiding division by zero, and ρ_{ref} is a reference BRDF, relative to which the mapping is applied. Like that in [NJR15], we choose the reference BRDF to be the median value for each $(\theta_h, \theta_d, \phi_d)$, over the entire MERL dataset.

3.2. Network Design

After BRDF data preprocessing, we design and train a deep autoencoder [HS06, Ben09] to explore the low-dimensional representation of input BRDFs. The autoencoder uses an encoder to reduce the dimensionality of the input data $\mathbf{X}$ and to extract a low-dimensional latent feature $\mathbf{Y}$ for each BRDF. Mathematically, we have

$$\mathbf{Y} = f_\theta(\mathbf{X}) \tag{2}$$

where f_θ denotes the encoder parameterized by θ. Then, a subsequent decoder network transforms the latent representation $\mathbf{Y}$ back into a high-dimensional output $\hat{\mathbf{X}}$ that is expected to be similar with the input, i.e.,

$$\hat{\mathbf{X}} = g_\theta(\mathbf{Y}). \tag{3}$$

The set of parameters θ of the encoder and decoder are learned simultaneously on the task of BRDF reconstruction.

Table 1: *The proposed autoencoder's architecture.*

Layer	Kernel	Stride	Resolution
Input			$540 \times 90 \times 90$
Conv2D	3×3	2	$256 \times 45 \times 45$
Conv2D	3×3	2	$128 \times 23 \times 23$
Conv2D	3×3	2	$64 \times 12 \times 12$
ResidualBlock	3×3	1	$64 \times 12 \times 12$
ResidualBlock	3×3	1	$64 \times 12 \times 12$
ResidualBlock	3×3	1	$64 \times 12 \times 12$
FC-10			
FC-$64 \times 12 \times 12$			
DeConv2D	3×3	2	$128 \times 24 \times 24$
DeConv2D	3×3	2	$256 \times 48 \times 48$
DeConv2D	4×4	2	$540 \times 90 \times 90$
Output			$540 \times 90 \times 90$

Table 1 illustrates the network architecture of our autoencoder. In the proposed network, three 2D convolutional layers (Conv2D) with kernel size 3×3 and stride 2 are used to downsample the input data. The resolution is halved after each convolutional layer. Three residual blocks (ResidualBlock) [HZRS16] are inserted to improve the performance of learning. Each residual block contains two convolutional layers, a leaky ReLU activation unit and a residual connection. These convolutional steps extend the receptive fields of the proposed network. To extract the low-dimensional latent vector, a fully connected layer (FC-10) is employed. To decode the latent vector, we expand this vector with another fully connected layer (FC-$64 \times 12 \times 12$), followed by three deconvolutional layers (Deconv2D).

The input of the autoencoder is the full data of a BRDF represented by reordered Rusinkiewicz half-difference angle coordinates $(\phi_d, \theta_h, \theta_d)$ and reshaped into $540 \times 90 \times 90$. In this way, we actually consider each measured BRDF as a sequence of image slices. Once trained, the stacked image slices can be perfectly recovered by our network. We choose such a BRDF parametrization approach because this makes all image slices quite similar except the invalid regions † as shown in the first row of Fig. 2. This allows us to vary

† Invalid entries represent those regions where either the incident direction or the viewing direction is below the horizon [Bur12]. These entries are set to zero during convolution.

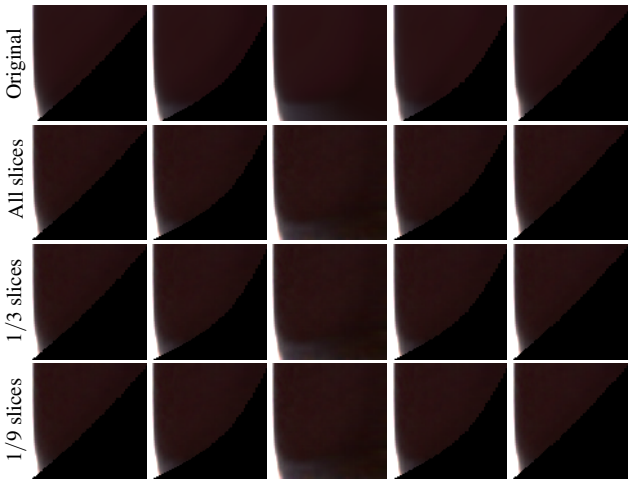

Figure 2: *Demonstration of original BRDF slices of* SPECULAR-VIOLET-PHENOLIC *and the reconstructed BRDF slices by our network with different numbers of input slices.*

the number of input slices. Even if some slices of a BRDF are missing, we can easily fill each missing slice with its nearest neighbor and the full BRDF data can still be recovered with high fidelity due to the high similarity of nearby slices. The subtle differences of image slices are only responsible for tiny features in the corresponding BRDF. In Fig. 2, we show the reconstructed slices of SPECULAR-VIOLET-PHENOLIC by our network with all, $1/3$ and $1/9$ original slices as the input, respectively [‡]. As seen, the visual differences are negligible, proving that our network is robust to incomplete or sparsely sampled BRDF data. Please refer to the supplemental material for more comparisons on synthesized images.

We implement and train our network using PyTorch. The network is optimized by the Adam optimizer [KB14] with a learning rate of 2×10^{-5} and mini-batches of size 32. The training converges after 1000 epochs which takes roughly 2 hours.

Dimensionality of the Latent Vector. The quality of reconstruction is affected by many factors. Here, we discuss the influence of the latent vector's dimensionality k. In general, the higher the dimensionality, the more information will be retained after reconstruction. This is evidenced in Fig. 3 where we show the reconstruction error (in terms of the Mean Squared Error) with respect to the latent vector's dimensionality. As seen from the trend of the red curve, an obvious decrease of the reconstruction error is observed as varying k from 5 to 10. After that, the error decreases steadily when k increases from 10 to 40. However, as k increases, the computational cost becomes large. Taking these factors into account, we generally set k as 10 without any specific tuning.

Loss Function. The loss function plays a key role in training a deep neural network. For our autoencoder, the goal is to recover the compressed high-dimensional BRDF data with minimal reconstruction error. To this end, we formulate the loss function as

$$\mathcal{L} = \sum_{\mathbf{X} \in \mathcal{D}_{\text{train}}} \| mask(g_\theta(f_\theta(\mathbf{X}))) - mask(\mathbf{X}) \| \qquad (4)$$

where $\mathbf{X}$ is the input BRDF data from our train dataset $\mathcal{D}_{\text{train}}$. Since some BRDF slices contain invalid values as shown in Fig. 2, we employ a *mask* function to return the index set of all valid BRDF data items. We observe that the *mask* function significantly improves the reconstruction quality since many entries in the original image slices are actually invalid. The norm $\| \cdot \|$ can be evaluated either under L_1 or L_2, resulting in L_1 loss and L_2 loss, respectively. We have compared L_1 loss and L_2 loss in designing the loss function. The differences are shown in Fig. 4. From the comparison we see that L_2 loss generally surpasses L_1 loss both qualitatively and quantitatively in our network. The network output generated by L_1 loss is slightly biased. The error metric we choose for quantitative comparison is Relative Absolute Error (RelAE) which is calculated in the image space.

[‡] Currently, the slices are equally spaced.

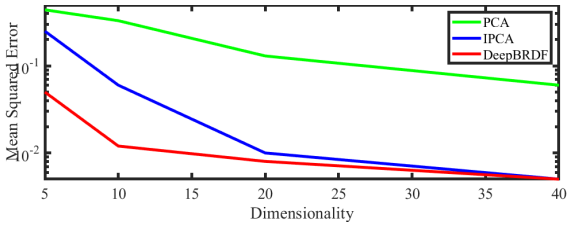

Figure 3: *Mean squared error (evaluated on the whole training dataset) as a function of the dimensionality of the underly subspace for traditional PCA, improved PCA (IPCA) [NJR15] and DeepBRDF, respectively. The proposed DeepBRDF clearly outperforms PCA-based strategies in BRDF data compression.*

Figure 4: *Comparison between L_1 loss and L_2 loss in designing the loss function. The corresponding Relative Absolute Error (RelAE) for each reconstructed result is reported.*

3.3. Geometric Interpretation

It is widely recognized that the high-dimensional BRDF data captured from real-world materials concentrate close to a nonlinear

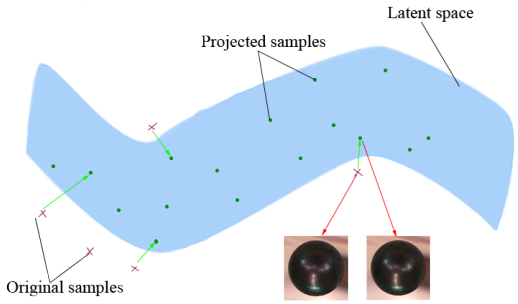

Figure 5: *Geometric interpretation of DeepBRDF. The proposed autoencoder projects all the original samples in the MERL BRDF dataset to a low-dimensional manifold latent space with minimal reconstruction error.*

low-dimensional manifold. The proposed deep autoencoder can be seen as a way to define and learn such a manifold, if we restrict the dimension of the bottleneck layer to be much smaller than the dimension of the original input. Fig. 5 illustrates that the proposed deep autoencoder tries to project the original BRDF data (crosses) in the MERL dataset to the latent low-dimensional manifold where these examples concentrate. One can think of the low-dimensional latent vector (dots) as a representation of the original BRDF which is well suited to capture the main variations in the data.

4. Quality Analysis

This section evaluates the quality of the propose DeepBRDF in BRDF compression by comparing with PCA-based strategies [MPBM03, NDM06]. Recall that PCA-based compression for the MERL dataset typically arranges all BRDFs as columns of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, where $n = 180 \times 90 \times 90 \times 3 = 4,374,000$ is the feature number, $m = 100$ is sample number. This matrix is subsequently decomposed using Singular Value Decomposition (SVD):

$$\mathbf{A} - \hat{\boldsymbol{\mu}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_{i=1}^{k} \boldsymbol{\sigma}_i \mathbf{u}_i \mathbf{v}_i^T \tag{5}$$

where k denotes the rank of the approximation which is also the dimension of the subspace. Each row of $\hat{\boldsymbol{\mu}} \in \mathbb{R}^{m \times n}$ is the mean over m rows of $\mathbf{A}$. The columns of $\mathbf{V} \in \mathbb{R}^{n \times k}$ are eigenvectors of the covariance $(\mathbf{A} - \hat{\boldsymbol{\mu}})^T (\mathbf{A} - \hat{\boldsymbol{\mu}})$, corresponding to the principal components of the data. Each diagonal element of $\boldsymbol{\Sigma} \in \mathbb{R}^{k \times k}$ is the variance explained by the corresponding principal component. To handle high dynamic range information, traditional PCA simply applies the natural logarithm to each element of $\mathbf{A}$. Recently, Nielsen et al. [NJR15] improved traditional PCA (i.e., IPCA) by applying a log-relative mapping to raw BRDF values. This results in higher reconstruction quality as compared in Fig. 6.

Fig. 6 reveals that synthesized images using BRDFs reconstructed by traditional PCA deviate greatly from the reference images even if the dimension of the underly subspace is high (e.g., 40D), due to the inherent linearity and high dynamic range of the original BRDF data. IPCA of Nielsen et al. [NJR15] significantly improves

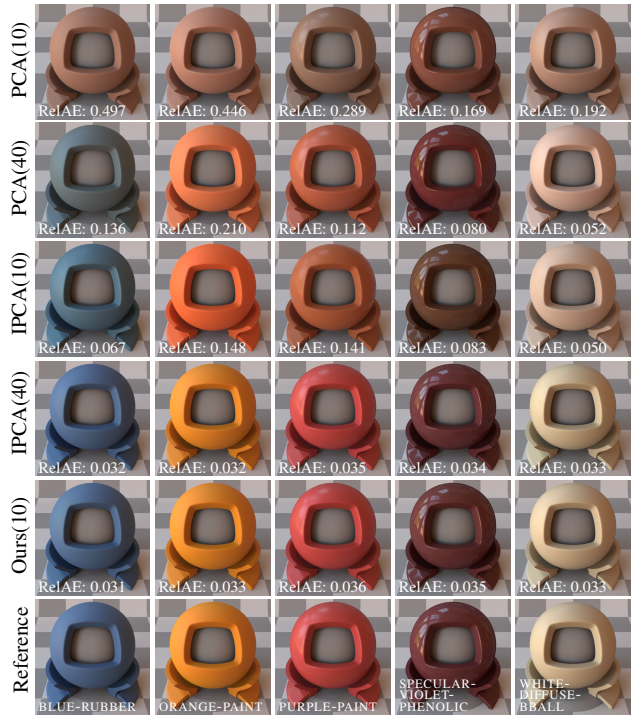

Figure 6: *Reconstruction quality comparison of our DeepBRDF against PCA and IPCA [NJR15]. For PCA and IPCA, the reconstruction error decreases as the dimensionality increases. However, we achieve high-quality reconstruction even if the dimension of the manifold is low (e.g., 10D). Quantitative evaluation in terms of RelAE is provided for each reconstructed result. Please refer to the supplemental material for more comparisons.*

traditional PCA both visually and in terms of RelAE. It converges quickly as the dimensionality increases and achieves satisfactory results in a 40D subspace. However, our method based on DeepBRDF can achieve high-quality reconstruction even if the underlying manifold lies in a low-dimensional subspace (e.g., 10D).

Fig. 7 reports all the reconstruction errors (in terms of RelAE) of the materials in the MERL dataset, as compared with two PCA-based methods. Clearly, our method outperforms IPCA on almost all materials when the dimensionality is equal. It achieves a similar reconstruction quality with IPCA in a 40D subspace. Fig. 3 also provides the comparison of averaged reconstruction errors (in terms of the Mean Squared Error) of the whole dataset.

Fig. 8 compares our method with that of Sun et al. [SJR18] which performs PCA for the diffuse part and the specular part of a BRDF, separately. Although it further improves IPCA, it is still inferior than our deep-learning-based solution, especially on the materials with changing color or complex lobes.

To show the generality and robustness of the proposed reconstruction method based on DeepBRDF, we validate it on BRDF data beyond what is found in the MERL dataset. In Fig. 9, the first three BRDFs are from the EPFL dataset [DJ18] and the last two are from [NJR15]. The comparisons reveal that our method still beats

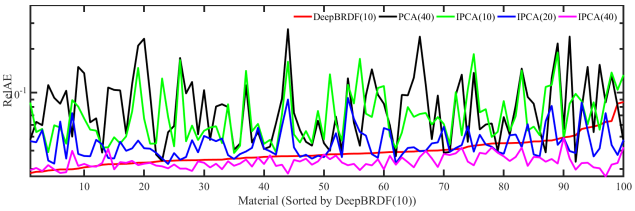

Figure 7: *Reconstruction error comparison of our DeepBRDF against PCA and IPCA [NJR15] with varying dimensions.*

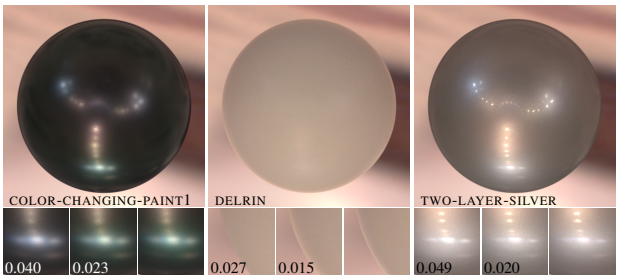

Figure 8: *Comparison between our DeepBRDF (10D) against the reconstruction method of Sun et al. [SJR18]. For each BRDF, we assign 1D for its diffuse part and 9D for its specular part in the method of Sun et al. [SJR18]. From left to right in each group of closeups, we compare the method of Sun et al. [SJR18], ours and the reference, with corresponding RelAE.*

IPCA both qualitatively and quantitatively. We also notice that the reconstruction errors are large for the EPFL dataset due to the fact that BRDFs in the EPFL dataset behave differently from that in the MERL dataset. We expect that including these BRDFs in training will make our DeepBRDF more expressive and significantly reduce the RelAE. Concerning timing performance, reconstructing the full data of a BRDF for IPCA requires 0.15 seconds while our method uses 0.3 seconds. Therefore, once trained, our DeepBRDF is able to faithfully reconstruct more measured BRDFs with high efficiency.

5. Applications and Results

In this section, we demonstrate two common applications benefitting from the proposed DeepBRDF.

5.1. BRDF Editing

We first show the capability of DeepBRDF in BRDF editing. Recall that BRDF editing is straightforward for analytic BRDFs but has not so far been easy to do for measured BRDFs [MPBM03, SGM*16, TKL*16]. However, once we get a low-dimensional representation of a material based on DeepBRDF, we can easily perform BRDF editing by modifying this low-dimensional vector. As demonstrated in Fig. 10, our DeepBRDF supports smooth and perceptually linear changes when interpolating between a very specular material and a diffuse one. In comparison, PCA-based strategies yield strange artifacts when the dimensionality is low and sudden changes in appearance even if the dimensionality is high.

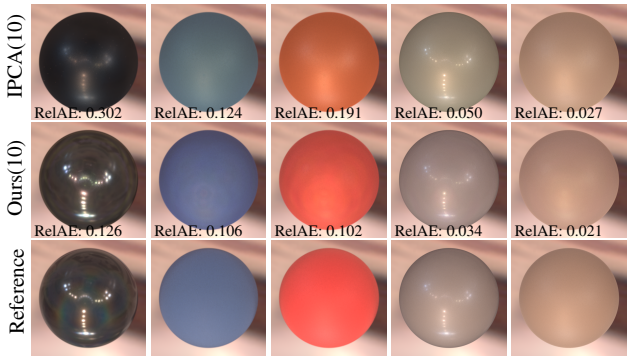

Figure 9: *Reconstruction quality comparison between our method and IPCA on measured BRDFs from the EPFL dataset [DJ18] (the left three columns) and [NJR15] (the right two columns), respectively.*

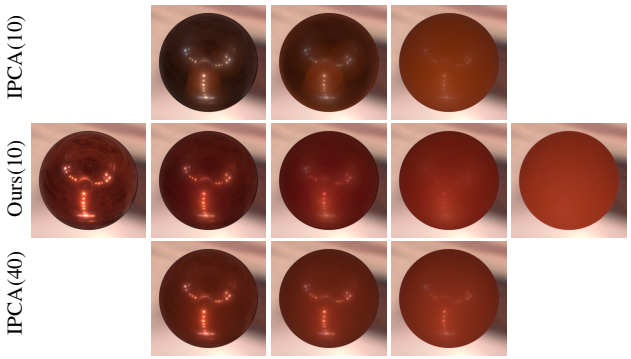

Figure 10: *Linear interpolation between* RED-METALLIC-PAINT *and* RED-FABRIC *using IPCA and our DeepBRDF, respectively.*

It should be noted that these latent vectors are not intuitive to the user and it is not possible for the user to directly change these vector values to achieve the desired editing effect. To alleviate this problem, we resort to a set of easy-to-understand and intuitive material properties that are convenient for BRDF editing. Then, we achieve the goal of BRDF editing by establishing a mapping of these attributes to the latent vector and adjusting these attributes instead of the latent vector. There are several different sets of attributes available for BRDF editing and we choose those proposed by Guo et al. [GGPL18] because they can alleviate the color shifting problem when editing the roughness, benefiting by a diffuse-specular separation.

The mapping between the editable attributes $\boldsymbol{\alpha}$ and the low-dimensional latent vector $\mathbf{Y}$ is formulated as

$$\mathbf{Y} = \Phi(\boldsymbol{\alpha}). \tag{6}$$

Here, $\boldsymbol{\alpha} \in \mathbb{R}^7$ contains our selected attributes. Currently, we choose three low-level attributes, i.e., diffuse albedo ($\boldsymbol{\alpha}_d \in \mathbb{R}^3$), specular albedo ($\boldsymbol{\alpha}_s \in \mathbb{R}^3$), and roughness factor ($g \in \mathbb{R}$). We use a Back Propagation (BP) regression network as a subnetwork to establish the relationship between $\mathbf{Y}$ and $\boldsymbol{\alpha}$. This subnetwork is actually a shallow fully connected neural network, as shown in Fig. 11. It

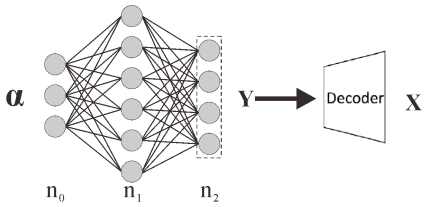

Figure 11: *The network architecture for BRDF editing based on DeepBRDF. A shallow fully connected neural network is adopted as a subnetwork to encode the mapping between the attribute vector* $\boldsymbol{\alpha}$ *and the latent vector* $\mathbf{Y}$*. The pre-trained decoder converts* $\mathbf{Y}$ *back to the full BRDF data* $\mathbf{X}$*.*

contains an input layer ($n_0 = 7$ neurons), a hidden layer ($n_1 = 100$ neurons) and an output layer ($n_2 = 10$ neurons). The pre-trained decoder is connected to the output layer and converts $\mathbf{Y}$ back to the full BRDF data $\mathbf{X}$.

Mathematically, this subnetwork is expressed as

$$\mathbf{Y} = \sum_{i=0}^{n_2} \mathbf{w}_i (R(\sum_{j=0}^{n_1} (\mathbf{w}_j \boldsymbol{\alpha} + \mathbf{b}_1)) + \mathbf{b}_2) \tag{7}$$

where $\mathbf{w}_i/\mathbf{w}_j$ represents the weight of each neuron, and $\mathbf{b}_1/\mathbf{b}_2$ is the bias. R is a leaky ReLU activation function. We train this subnetwork independently using the extracted latent vectors of 100 MERL BRDFs and the corresponding attributes provided by Guo et al. [GGPL18]. The loss function is simply $\|\hat{\mathbf{Y}} - \mathbf{Y}\|/\|\mathbf{Y}\|$ in which $\hat{\mathbf{Y}}$ is the network output.

Since the selection of attributes $\boldsymbol{\alpha}$ and the selection of underlying low-dimensional vectors $\mathbf{Y}$ are uncorrelated, we can certainly choose other attributes such as those proposed by Serrano et al. [SGM*16] as the input $\boldsymbol{\alpha}$. We only have to retrain the fully connected neural network and fix the decoder. Training this subnetwork is quite efficient since it is shallow.

Editing Albedos. We have two types of albedos in our selected attributes and they can be edited independently without influencing each other. Since the albedo is encoded in the RGB color space, we directly change each channel to achieve the desired color effect. Editing the diffuse albedo is shown in Fig. 12. Notice that the color and the shape of the highlights do not change when we alter the diffuse albedo. In Fig. 1, we show a scene contains several edited materials by changing the diffuse albedo. This scene only contains two original BRDFs (RED-PLASTIC and SILVER-METALLIC-PAINT) from the MERL dataset. A large number of physically plausible BRDFs can be generated using the proposed editing framework.

Adjusting Roughness. Roughness determines the shape of the highlight. We can achieve highlight control by adjusting the roughness without changing the color of the surface. Fig. 13 lists a series of different rendering results of decreased roughness. We compare our method against the method of Serrano et al. [SGM*16] which provides thirteen high-level attributes to control the material appearances. However, there exists strong correlations between some of the attributes, e.g., glossy and metallic-like are highly correlated. This may cause confusion for the user. In Fig. 13, we use their

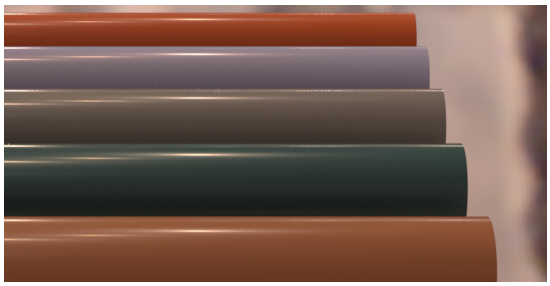

Figure 12: *Editing the diffuse albedo of* GRAY-PLASTIC *(cylinder in the middle) while keeping other attributes unchanged.*

Figure 13: *Editing the roughness of* RED-PLASTIC *in the left-most column. The roughness decreases from left to right. The first row contains our editing results while the second row contains the editing results of Serrano et al. [SGM*16].*

method to edit the roughness of RED-PLASTIC. As seen, there is obvious color shift during editing using the method of Serrano et al. while our method is free from this.

We also make a comparison between IPCA [NJR15] and DeepBRDF as a low-dimensional representation in BRDF editing. The results of increasing the roughness of SPECULAR-YELLOW-PHENOLIC are provided in Fig. 14. Recall that the IPCA-based representation with a low dimensionality (e.g., 10D) can not preserve the appearance quite well. Therefore, the diffuse color may change during editing. Despite that, we observe that BRDF editing based on IPCA is unable to guarantee smooth transitions between BRDFs. The perception of glossiness may fluctuate strangely as shown in the bottom row of Fig. 14. In comparison, our method based on DeepBRDF provides both smooth transitions and high physical plausibility.

5.2. Single Image BRDF Recovery

Recovering the BRDF from a single input image is an ill-posed inverse problem since infinitely many combinations of material reflectance, illumination and geometries can create the same image. Most previous methods assume that the underlying BRDF model is analytical such that only a few parameters (e.g., roughness) are required to estimate. Unfortunately, analytical models limit reflectance behaviors that can be predicted. Therefore, it is more accurate to recover a tabulated data-driven BRDF model from an input image. Thanks to DeepBRDF, we are able to build a connection

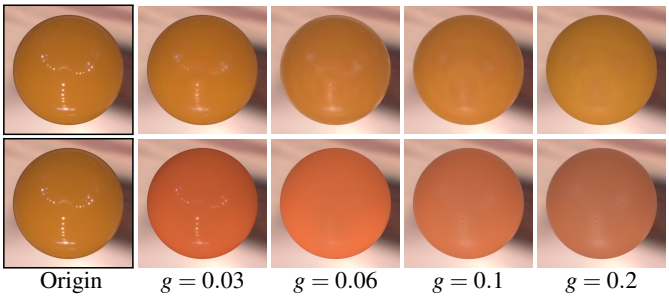

Figure 14: *Editing the roughness of* SPECULAR-YELLOW-PHENOLIC *with IPCA-based representation (bottom row) and DeepBRDF-based representation (top row), respectively. The roughness increases from left to right. Despite the color shift, increasing the roughness with IPCA-based representation fails to produce smoothing transitions across BRDFs.*

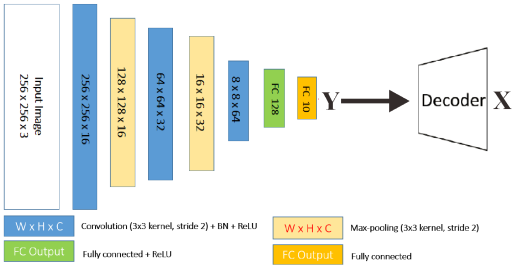

Figure 15: *The network architecture for recovering BRDF data from a single input image. A new CNN mapping the input image to the latent space of DeepBRDF is connected to the pre-trained decoder. The decoder decompresses the latent vector and generates the full data of a BRDF.*

between the full data of a material and the input image, with DeepBRDF serving as the "bridge".

Once the previous autoencoder has learned a latent representation for measured BRDFs, we train another CNN which learns to map an input image to the latent space of this autoencoder. This CNN takes as input a 256×256 color image which is then passed through a series of three convolutional layers, two max-pooling layers and two fully connected layers. The network architecture of this CNN is shown in Fig. 15. Max-pooling layers are used to preserve more highlights details and retain color information. L_1 loss is used as the loss function. During training, we use Adam optimizer with a batch size of 64 and a learning rate of 1×10^{-3}. The model converges after 300 epochs.

Training Dataset. To train the above CNN, we generate 10000 synthetic images using the Mitsuba renderer [Jak10]. Each image only contains one virtual object captured at different viewpoints. The viewpoints have been chosen to cover various shapes of highlights. The measured materials are all from the MERL BRDF dataset [MPBM03]. We render each object under sunsky lighting to ensure that the rendered object contains less environmental information. Some training examples are provided in the supplemental material.

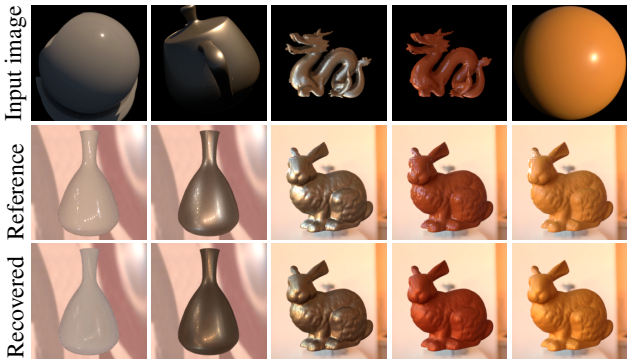

Figure 16: *BRDF recovery results of synthetic images. The first row shows the input images. The remained two rows show the rendered images using the ground-truth materials and recovered materials, respectively.*

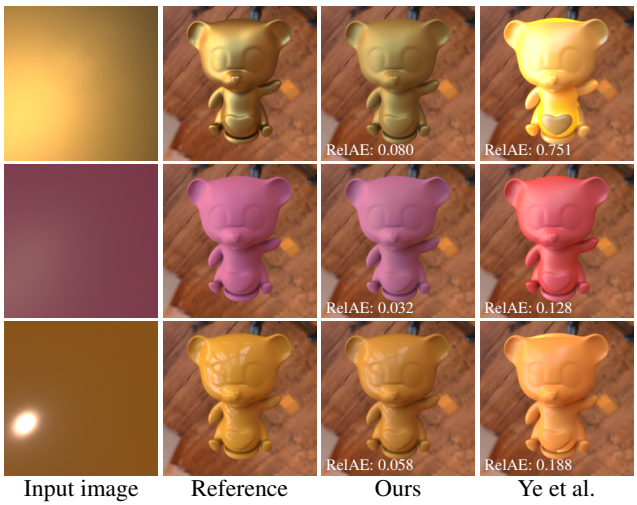

Figure 17: *Comparison with the method of Ye et al. [YLD*18] in homogeneous BRDF recovery. For a fair comparison, the input images are all rendered with a single plane.*

Evaluation Results. We have conducted experiments on both synthetic images and real-world data. Fig. 16 shows some samples on synthetic images. Note that these images have not been used for training. As seen from the pair-wise comparison with reference images generated by ground-truth materials, our method can recover the BRDF data quite well. The recovered BRDFs preserve most of the reflectance behaviors such as highlights and diffuse colors of the intrinsic BRDF data. Even if the highlight is subtle in the image, our method can sill reproduce such high-frequency information with high fidelity. Fig. 17 compares our method with the method of Ye et al. [YLD*18] in recovering homogeneous BRDFs. Note that the method of Ye et al. [YLD*18] can only handle planar material samples. Therefore, for a fair comparison, we only use images rendered with a plane in this figure. The results tell that our method

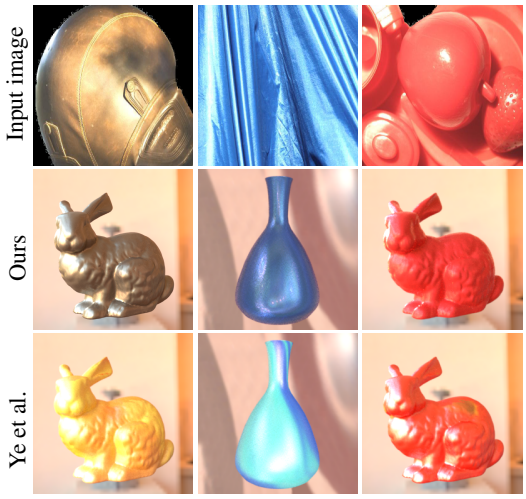

Figure 18: *BRDF recovery results for real-world images.*

outperforms the method of Ye et al. [YLD*18] both qualitatively and quantitatively. The main reason is that their method uses an analytical Ward model [War92] to encoder surface reflectance. Although simple, this will limit the range of appearances that can be recovered. In contrast, our method based on measured BRDFs from real-world materials can handle a much wider range of appearances, leading to higher recovery quality. We finally compare our method with the method of Ye et al. [YLD*18] in recovering BRDFs from real-world images. These input images in Fig. 18 are taken from the Flickr Material Database (FMD) [SRA09]. Again, our method performs better than its competitor and yields appearances that are consistent with those in the input images. Currently, our method achieves good predictions when the geometry is relatively simple and the images well capture the highlights.

6. Conclusion and Future Work

We have presented DeepBRDF, a deep-learning-based representation for measured BRDFs from real-world materials. With DeepBRDF, we are able to encode high-dimensional tabulated BRDFs in a nonlinear low-dimensional subspace without suffering from large reconstruction error. This new representation has potential benefits for many graphical and vision applications manipulating measured BRDFs and we have demonstrated two common ones in this paper. We have shown that the proposed DeepBRDF allows efficient BRDF editing with smooth transitions and high physical plausibility via building a relationship between some perceptual attributes and the latent vector. We have also validated the superiority of DeepBRDF in single image BRDF recovery by introducing a new CNN learning the mapping from an input color image to the latent vector of a BRDF, and shown that DeepBRDF clearly outperforms previous methods relying on analytical BRDF models. For future work, we will exploit DeepBRDF in more applications, such as gamut mapping [SSGM17, GGPL18] and material perception [LMS*19], with DeepBRDF serving as a material appearance similarity measure of data-driven BRDF models.

Acknowledgements

We would like to thank the reviewers for their valuable feedback. This work was partially supported by the National Key Research and Development Program of China (No. 2018YFB1004901) and NSFC (No. 61972194).

References

[BAOR06] BEN-ARTZI A., OVERBECK R., RAMAMOORTHI R.: Real-time brdf editing in complex lighting. In *ACM SIGGRAPH 2006 Papers* (2006), SIGGRAPH '06, pp. 945–954. 2

[Ben09] BENGIO Y.: Learning deep architectures for ai. *Foundations and trends in Machine Learning 2* (2009), 1–127. 2, 3

[BM15] BARRON J. T., MALIK J.: Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 8 (2015), 1670–1687. 3

[BÖK11] BILGILI A., ÖZTÜRK A., KURT M.: A general BRDF representation based on tensor decomposition. *Comput. Graph. Forum 30*, 8 (2011), 2427–2439. 2

[BSH12] BAGHER M. M., SOLER C., HOLZSCHUCH N.: Accurate fitting of measured reflectances using a shifted gamma micro-facet distribution. *Comput. Graph. Forum 31*, 4 (June 2012), 1509–1518. 2

[Bur12] BURLEY B.: Physically based shading at Disney. In *ACM SIGGRAPH 2012 Course Notes: Practical Physically-Based Shading in Film and Gram Production* (2012), SIGGRAPH '12 Courses. 2, 3

[CPK06] COLBERT M., PATTANAIK S., KRIVANEK J.: Brdf-shop: creating physically correct bidirectional reflectance distribution functions. *IEEE Computer Graphics and Applications 26*, 1 (2006), 30–36. 2

[DJ18] DUPUY J., JAKOB W.: An adaptive parameterization for efficient material acquisition and rendering. *Transactions on Graphics (Proceedings of SIGGRAPH Asia) 37*, 6 (Nov. 2018), 274:1–274:18. 1, 2, 5, 6

[DRS08] DORSEY J., RUSHMEIER H., SILLION F.: *Digital Modeling of Material Appearance*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008. 1

[FV14] FILIP J., VÁVRA R.: Template-based sampling of anisotropic BRDFs. *Computer Graphics Forum 33*, 7 (2014), 91–99. 1, 2

[GGG*16] GUARNERA D., GUARNERA G. C., GHOSH A., DENK C., GLENCROSS M.: BRDF representation and acquisition. *Comput. Graph. Forum* (2016). 1, 2

[GGPL18] GUO J., GUO Y., PAN J., LU W.: BRDF analysis with directional statistics and its applications. *IEEE Transactions on Visualization and Computer Graphics PP* (10 2018). 2, 6, 7, 9

[GLD*19] GAO D., LI X., DONG Y., PEERS P., XU K., TONG X.: Deep inverse rendering for high resolution SVBRDF estimation from an arbitrary number of images. *ACM Transactions on Graphics (SIGGRAPH 2019) 37*, 4 (July 2019). 2, 3

[HS06] HINTON G. E., SALAKHUTDINOV R. R.: Reducing the dimensionality of data with neural networks. *Science 313*, 5786 (2006), 504–507. 2, 3

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778. 3

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 8

[KB14] KINGMA D., BA J.: Adam: A method for stochastic optimization. *International Conference on Learning Representations* (12 2014). 4

[KCW*18] KANG K., CHEN Z., WANG J., ZHOU K., WU H.: Efficient reflectance capture using an autoencoder. *ACM Trans. Graph. 37*, 4 (July 2018), 127:1–127:10. 2, 3

[KGT*17] KIM K., GU J., TYREE S., MOLCHANOV P., NIESSNER M., KAUTZ J.: A lightweight approach for on-the-fly reflectance estimation. In *The IEEE International Conference on Computer Vision (ICCV)* (Oct 2017). 2

[KWKT15] KULKARNI T. D., WHITNEY W. F., KOHLI P., TENENBAUM J. B.: Deep convolutional inverse graphics network. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (2015), NIPS'15, pp. 2539–2547. 3

[LBAD*06] LAWRENCE J., BEN-ARTZI A., DECORO C., MATUSIK W., PFISTER H., RAMAMOORTHI R., RUSINKIEWICZ S.: Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25*, 3 (July 2006). 2

[LCY*17] LIU G., CEYLAN D., YUMER E., YANG J., LIEN J.-M.: Material editing using a physically based rendering network. In *The IEEE International Conference on Computer Vision (ICCV)* (Oct 2017). 3

[LDPT17] LI X., DONG Y., PEERS P., TONG X.: Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Trans. Graph. 36*, 4 (July 2017), 45:1–45:11. 2, 3

[LKYU12] LÖW J., KRONANDER J., YNNERMAN A., UNGER J.: BRDF models for accurate and efficient rendering of glossy surfaces. *ACM Trans. Graph. 31*, 1 (Feb. 2012), 9:1–9:14. 2

[LMS*19] LAGUNAS M., MALPICA S., SERRANO A., GARCES E., GUTIERREZ D., MASIA B.: A similarity measure for material appearance. *ACM Transactions on Graphics (SIGGRAPH 2019) 38*, 4 (2019). 2, 9

[LRR04] LAWRENCE J., RUSINKIEWICZ S., RAMAMOORTHI R.: Efficient brdf importance sampling using a factored representation. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 496–505. 2

[LXR*18] LI Z., XU Z., RAMAMOORTHI R., SUNKAVALLI K., CHANDRAKER M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph. 37*, 6 (Dec. 2018), 269:1–269:11. 2, 3

[MAA01] MCCOOL M. D., ANG J., AHMAD A.: Homomorphic factorization of BRDFs for high-performance rendering. In *Proc. SIGGRAPH 2001* (2001), pp. 171–178. 2

[MMZ*18] MEKA A., MAXIMOV M., ZOLLHOEFER M., CHATTERJEE A., SEIDEL H.-P., RICHARDT C., THEOBALT C.: Lime: Live intrinsic material estimation. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (June 2018). 2, 3

[MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3 (July 2003), 759–769. 1, 2, 3, 5, 6, 8

[NDM05] NGAN A., DURAND F., MATUSIK W.: Experimental analysis of BRDF models. In *Proc. Eurographics Conference on Rendering Techniques* (2005), EGSR '05, pp. 117–126. 2

[NDM06] NGAN A., DURAND F., MATUSIK W.: Image-driven navigation of analytical BRDF models. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques* (2006), EGSR '06, pp. 399–407. 2, 5

[NJR15] NIELSEN J. B., JENSEN H. W., RAMAMOORTHI R.: On optimal, minimal BRDF sampling for reflectance acquisition. *ACM Transactions on Graphics (TOG) 34*, 6 (November 2015), 186:1–186:11. 2, 3, 4, 5, 6, 7

[NKLN10] NGUYEN C. H., KYUNG M.-H., LEE J.-H., NAM S.-W.: A PCA Decomposition for Real-time BRDF Editing and Relighting with Global Illumination. *Computer Graphics Forum* (2010). 2

[NMXY15] NARIHIRA T., MAIRE M., X. YU S.: Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *2015 IEEE International Conference on Computer Vision (ICCV)* (12 2015), pp. 2992–2992. 3

[RBMS17] RIBARDIÈRE M., BRINGIER B., MENEVEAUX D., SIMONOT L.: Std: Student's t-distribution of slopes for microfacet based bsdfs. *Comput. Graph. Forum 36*, 2 (2017), 421–429. 2

[RK10] RUITERS R., KLEIN R.: *A compact and editable representation for measured BRDFs*. Tech. Rep. CG-2010-1, University of Bonn, Dec. 2010. 2

[RRF*16] REMATAS K., RITSCHEL T., FRITZ M., GAVVES E., TUYTELAARS T.: Deep reflectance maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). 3

[Rus01] RUSINKIEWICZ S.: A new change of variables for efficient BRDF representation. *EuroGraphics Symposium on Rendering* (10 2001). 3

[RZ10] ROMEIRO F., ZICKLER T.: Blind reflectometry. In *Proceedings of the 11th European Conference on Computer Vision: Part I* (2010), ECCV'10, pp. 45–58. 3

[SGM*16] SERRANO A., GUTIERREZ D., MYSZKOWSKI K., SEIDEL H.-P., MASIA B.: An intuitive control space for material appearance. *ACM Trans. Graph. 35*, 6 (Nov. 2016), 186:1–186:12. 2, 3, 6, 7

[SJR18] SUN T., JENSEN H. W., RAMAMOORTHI R.: Connecting measured brdfs to analytic brdfs by data-driven diffuse-specular separation. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 273. 2, 5, 6

[SPN*16] SCHMIDT T.-W., PELLACINI F., NOWROUZEZAHRAI D., JAROSZ W., DACHSBACHER C.: State of the art in artistic editing of appearance, lighting and material. *Computer Graphics Forum 35*, 1 (2016), 216–233. 2

[SRA09] SHARAN L., ROSENHOLTZ R., ADELSON E. H.: Material perception: What can you see in a brief glance? *Journal of Vision 9*, 8 (2009). 9

[SSGM17] SUN T., SERRANO A., GUTIERREZ D., MASIA B.: Attribute-preserving gamut mapping of measured brdfs. *Computer Graphics Forum 36*, 4 (2017). 9

[SSN18] SOLER C., SUBR K., NOWROUZEZAHRAI D.: A Versatile Parameterization for Measured Material Manifolds. *Computer Graphics Forum 37*, 2 (Apr. 2018), 135–144. 2

[SZC*07] SUN X., ZHOU K., CHEN Y., LIN S., SHI J., GUO B.: Interactive relighting with dynamic brdfs. *ACM Trans. Graph. 26*, 3 (July 2007). 2

[TKL*16] TSIRIKOGLOU A., KRONANDER J., LARSSON P., TONGBUASIRILAI T., GARDNER A., UNGER J.: Differential appearance editing for measured brdfs. In *ACM SIGGRAPH 2016 Talks* (2016). 2, 6

[TSH12] TANG Y., SALAKHUTDINOV R., HINTON G.: Deep lambertian networks. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012* (06 2012). 3

[WAKB09] WILLS J., AGARWAL S., KRIEGMAN D., BELONGIE S.: Toward a perceptual space for gloss. *ACM Trans. Graph. 28*, 4 (Sept. 2009), 103:1–103:15. 2

[War92] WARD G. J.: Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph. 26*, 2 (July 1992), 265–272. 9

[XNY*16] XU Z., NIELSEN J. B., YU J., JENSEN H. W., RAMAMOORTHI R.: Minimal BRDF sampling for two-shot near-field reflectance acquisition. *ACM Trans. Graph. 35*, 6 (Nov. 2016), 188:1–188:12. 2

[YLD*18] YE W., LI X., DONG Y., PEERS P., TONG X.: Single image surface appearance modeling with self-augmented cnns and inexact supervision. *Computer Graphics Forum 37* (10 2018), 201–211. 2, 3, 8, 9

[ZFWW18] ZSOLNAI-FEHÉR K., WONKA P., WIMMER M.: Gaussian material synthesis. *ACM Trans. Graph. 37*, 4 (July 2018), 76:1–76:14. 2

[ZKE15] ZHOU T., KRÄHENBÜHL P., EFROS A. A.: Learning data-driven reflectance priors for intrinsic image decomposition. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 3469–3477. 3

[ZREB06] ZICKLER T., RAMAMOORTHI R., ENRIQUE S., BELHUMEUR P. N.: Reflectance sharing: Predicting appearance from a sparse set of images of a known shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 8 (Aug 2006), 1287–1302. 3