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-
reactions 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, SGM*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 [D18].

**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. Matsukis 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 [LR04, 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-
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 $X$ and to extract a low-dimensional latent feature $Y$ for each BRDF. Mathematically, we have

$$Y = f_0(X)$$  \hspace{1cm} (2)

where $f_0$ denotes the encoder parameterized by $\theta$. Then, a subsequent decoder network transforms the latent representation $Y$ back into a high-dimensional output $\hat{X}$ that is expected to be similar with the input, i.e.,

$$\hat{X} = g_0(Y)$$  \hspace{1cm} (3)

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

Table 1 illustrates the network architecture of our autoencoder. In the proposed network, three 2D convolutional layers (Conv2D) with kernel size $3 \times 3$ 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 network.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Kernel</th>
<th>Stride</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>-</td>
<td>-</td>
<td>$540 \times 90 \times 90$</td>
</tr>
<tr>
<td>Conv2D</td>
<td>$3 \times 3$</td>
<td>2</td>
<td>$256 \times 45 \times 45$</td>
</tr>
<tr>
<td>Conv2D</td>
<td>$3 \times 3$</td>
<td>2</td>
<td>$128 \times 23 \times 23$</td>
</tr>
<tr>
<td>Conv2D</td>
<td>$3 \times 3$</td>
<td>2</td>
<td>$64 \times 12 \times 12$</td>
</tr>
<tr>
<td>ResidualBlock</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>$64 \times 12 \times 12$</td>
</tr>
<tr>
<td>ResidualBlock</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>$64 \times 12 \times 12$</td>
</tr>
<tr>
<td>ResidualBlock</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>$64 \times 12 \times 12$</td>
</tr>
<tr>
<td>DeConv2D</td>
<td>$3 \times 3$</td>
<td>2</td>
<td>$256 \times 48 \times 48$</td>
</tr>
<tr>
<td>DeConv2D</td>
<td>$4 \times 4$</td>
<td>2</td>
<td>$540 \times 90 \times 90$</td>
</tr>
<tr>
<td>Output</td>
<td>-</td>
<td>-</td>
<td>$540 \times 90 \times 90$</td>
</tr>
</tbody>
</table>

The proposed autoencoder's architecture.

\[ \text{Table 1: The proposed autoencoder's architecture.} \]

\[ \text{Layer} \quad \text{Kernel} \quad \text{Stride} \quad \text{Resolution} \]
\[ \text{Input} \quad 540 \times 90 \times 90 \]
\[ \text{Conv2D} \quad 3 \times 3 \quad 2 \quad 256 \times 45 \times 45 \]
\[ \text{Conv2D} \quad 3 \times 3 \quad 2 \quad 128 \times 23 \times 23 \]
\[ \text{Conv2D} \quad 3 \times 3 \quad 2 \quad 64 \times 12 \times 12 \]
\[ \text{ResidualBlock} \quad 3 \times 3 \quad 1 \quad 64 \times 12 \times 12 \]
\[ \text{ResidualBlock} \quad 3 \times 3 \quad 1 \quad 64 \times 12 \times 12 \]
\[ \text{ResidualBlock} \quad 3 \times 3 \quad 1 \quad 64 \times 12 \times 12 \]
\[ \text{DeConv2D} \quad 3 \times 3 \quad 2 \quad 256 \times 48 \times 48 \]
\[ \text{DeConv2D} \quad 4 \times 4 \quad 2 \quad 540 \times 90 \times 90 \]
\[ \text{Output} \quad 540 \times 90 \times 90 \]
Currently, the slices are equally spaced.

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 $2 \times 10^{-5}$ 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

$$
\mathcal{L} = \sum_{X \in \mathcal{D}_{\text{train}}} \| \text{mask}(g_{\theta}(f_{\theta}(X))) - \text{mask}(X) \|_2 
$$

(4)

where $X$ 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 $\| \|$ 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 corresponding Relative Absolute Error (RelAE) for each reconstructed result is reported.

**Geometric Interpretation**

It is widely recognized that the high-dimensional BRDF data captured from real-world materials concentrate close to a nonlinear
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 $A \in \mathbb{R}^{n \times m}$, where $n = 180 \times 90 \times 3 = 4,374,000$ is the feature number, $m = 100$ is sample number. This matrix is subsequently decomposed using Singular Value Decomposition (SVD):

$$A - \bar{\mu} = U\Sigma V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$  \hspace{1cm} (5)

where $k$ denotes the rank of the approximation which is also the dimension of the subspace. Each row of $\bar{\mu} \in \mathbb{R}^{n \times 1}$ is the mean over $m$ rows of $A$. The columns of $V \in \mathbb{R}^{n \times k}$ are eigenvectors of the covariance $(A - \bar{\mu})(A - \bar{\mu})^T$, corresponding to the principal components of the data. Each diagonal element of $\Sigma \in \mathbb{R}^{k \times k}$ 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 $A$. 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 underlying 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 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). Quantitative evaluation in terms of RelAE is provided for each reconstructed result. Please refer to the supplemental material for more comparisons.

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 to 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
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, SGM16,TKL16]. 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.

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 $\mathbf{a}$ and the low-dimensional latent vector $\mathbf{Y}$ is formulated as

$$\mathbf{Y} = \Phi(\mathbf{a}).$$

Here, $\mathbf{a} \in \mathbb{R}^7$ contains our selected attributes. Currently, we choose three low-level attributes, i.e., diffuse albedo ($\mathbf{a}_d \in \mathbb{R}^3$), specular albedo ($\mathbf{a}_s \in \mathbb{R}^3$), 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 $\mathbf{a}$. This subnetwork is actually a shallow fully connected neural network, as shown in Fig. 11. It...
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 Y back to the full BRDF data X.

Mathematically, this subnetwork is expressed as

\[ Y = \sum_{i=0}^{n_0} w_i (R(\sum_{j=0}^{n_1} (w_j \alpha + b_1)) + b_2) \]

where \( w_i/w_j \) represents the weight of each neuron, and \( b_1/b_2 \) 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 \( \| Y - \hat{Y} \| / \| Y \| \) in which \( \hat{Y} \) is the network output.

Since the selection of attributes \( \alpha \) and the selection of underlying low-dimensional vectors Y are uncorrelated, we can certainly choose other attributes such as those proposed by Serrano et al. [SGM*16] as the input \( \alpha \). 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 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) cannot 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

---

© 2020 The Author(s)
Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.
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 DeepBRDF. 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 $1 \times 10^{-3}$. 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.

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 still 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
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


[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


[NMB\textsuperscript{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


[L\texttextsuperscript{R04}]


[MPBM\textsuperscript{03}] Matusik W., Pfister H., Brand M., McCullin L.: A data-driven reflectance model. ACM Transactions on Graphics 22, 3 (July 2003), 759–769. 1, 2, 3, 5, 6, 8


[NJR\textsuperscript{15}] 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


[R\textsuperscript{RF}]

[S\textsuperscript{R01}] Rusinkiewicz S.: A new change of variables for efficient BRDF representation. EuroGraphics Symposium on Rendering (10 2001). 3


[S\textsuperscript{G\textsuperscript{M}}]

[S\textsuperscript{S\textsuperscript{R}16}] Serrano A., Gutierrrez 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

[S\textsuperscript{IR\textsuperscript{18}}] Sun T., Jensen H. W., Ramamoorthi R.: Connecting measured brdfs to analytic brdfs by data-driven diffuse-spectral separation. ACM Transactions on Graphics (TOG) 37, 6 (2018), 273. 2, 5, 6


[S\textsuperscript{S\textsuperscript{N18}}] Soler C., Subb K., Nowrouzezahrai D.: A Versatile Parameterization for Measured Material Manifolds. Computer Graphics Forum 37, 2 (Apr. 2018), 135–144. 2

[S\textsuperscript{SZ\textsuperscript{C}}]

[W\textsuperscript{AK\textsuperscript{B09}}] Wills J., Agarwal S., Kriegman D., Belongie S.: Toward a perceptual space for gloss. ACM Trans. Graph. 26, 3 (July 2007). 2


[W\textsuperscript{AK\textsuperscript{B90}}]

[Z\textsuperscript{F\textsuperscript{W\textsuperscript{18}}}]

[Z\textsuperscript{E\textsuperscript{K15}]}

[Z\textsuperscript{Y\textsuperscript{L}}]

[Z\textsuperscript{E\textsuperscript{B06}]}

© 2020 The Author(s)

Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.