



A Sparse Non-parametric BRDF Model

TANABOON TONGBUASIRILAI, Linköping University, Sweden

JONAS UNGER, Linköping University, Sweden

CHRISTINE GUILLEMOT, INRIA, France

EHSAN MIANDJI, Linköping University, Sweden

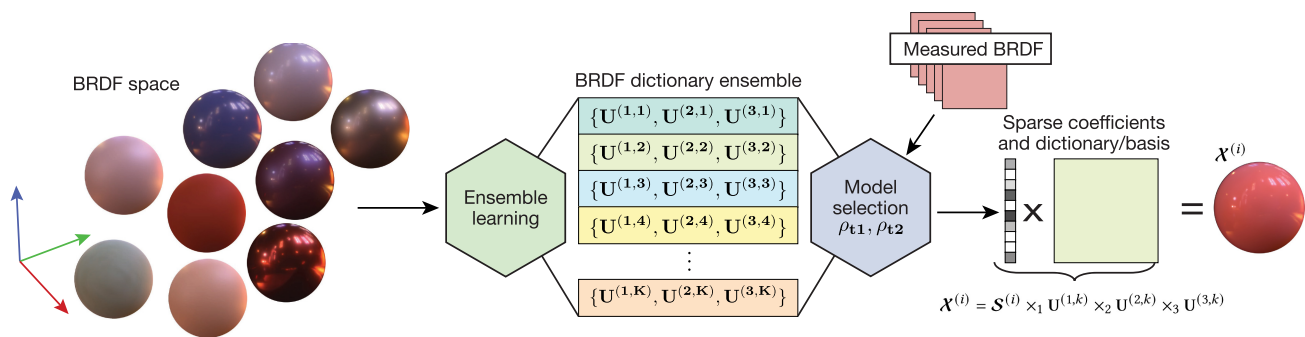


Fig. 1. Overview of the proposed framework for learning accurate representations and sparse data-driven BRDF models through analysis of the space of BRDFs. The BRDF dictionary ensemble is trained once and can accurately represent a wide range of previously unseen materials.

This paper presents a novel sparse non-parametric Bidirectional Reflectance Distribution Function (BRDF) model derived using a machine learning approach to represent the space of possible BRDFs using a set of multidimensional sub-spaces, or dictionaries. By training the dictionaries under a sparsity constraint, the model guarantees high-quality representations with minimal storage requirements and an inherent clustering of the BRDF-space. The model can be trained once and then reused to represent a wide variety of measured BRDFs. Moreover, the proposed method is flexible to incorporate new unobserved data sets, parameterizations, and transformations. In addition, we show that any two, or more, BRDFs can be smoothly interpolated in the coefficient space of the model rather than the significantly higher-dimensional BRDF space. The proposed sparse BRDF model is evaluated using the MERL, DTU, and RGL-EPFL BRDF databases. Experimental results show that the proposed approach results in about 9.75dB higher signal-to-noise ratio on average for rendered images as compared to current state-of-the-art models.

CCS Concepts: • **Computing methodologies** → **Reflectance modeling**; **Machine learning approaches**;

This work was funded through Knut and Alice Wallenberg Foundation (KAW), Wallenberg Autonomous Systems and Software Program (WASP), and the strategic research environment ELLIIT. Additionally, a part of this work was funded by EU H2020 Research, and Innovation Programme under Grant No. 694122 (ERC advanced Grant CLIM).

Authors' addresses: T. Tongbuasirilai, J. Unger, and E. Miandji, Linköping University, Linköping, 58183, Sweden; emails: {tanaboon.tongbuasirilai, jonas.unger, ehsan.miandji}@liu.se; C. Guillemot, INRIA, Rennes, France; email: christine.guillemot@inria.fr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://doi.org/10.1145/3533427).

© 2022 Association for Computing Machinery.

0730-0301/2022/10-ART181 \$15.00

<https://doi.org/10.1145/3533427>

Additional Key Words and Phrases: Rendering, reflectance and shading models, machine learning, dictionary learning, non-parametric BRDF model, BRDF interpolation

ACM Reference format:

Tanaboon Tongbuasirilai, Jonas Unger, Christine Guillemot, and Ehsan Miandji. 2022. A Sparse Non-parametric BRDF Model. *ACM Trans. Graph.* 41, 5, Article 181 (October 2022), 18 pages.

<https://doi.org/10.1145/3533427>

1 INTRODUCTION

The bidirectional reflectance distribution function [Nicodemus et al. 1977] describes how light scatters at the surfaces of a scene, depending on their material characteristics. The **Bidirectional Reflectance Distribution Function (BRDF)** is a 4D function parameterized by the incident and exitant scattering angles and can be described using either parametric models [Ashikhmin and Shirley 2000; Blinn 1977; Cook and Torrance 1982; Löw et al. 2012; Walter et al. 2007] or data-driven models [Bagher et al. 2016; Bilgili et al. 2011; Jakob et al. 2014; Lawrence et al. 2004; Sun et al. 2018; Tongbuasirilai et al. 2019]. Parametric models present great artistic freedom and the possibility to interactively tweak parameters to achieve the desired look and feel. However, most analytical models are not designed for efficient and accurate representation of the scattering properties of measured real-world materials. Data-driven models, however, enable the use of measured BRDFs and real-world materials directly in the rendering pipeline, and are commonly used in computer vision applications [Romeiro et al. 2008]. Data-driven models can represent BRDFs in many different ways. Iterative-factored representations approximate BRDFs with multiple low-rank components [Bilgili et al. 2011; Lawrence et al. 2004; Tongbuasirilai et al. 2019], while hybrid analytical data-driven models [Bagher et al. 2016; Sun et al. 2018] rely on

non-parametric components or basis functions computed using specific weighting and optimization schemes.

The efficiency, or performance, of a non-parametric model is typically measured in terms of the number of variables/coefficients required to represent a BRDF at a given quality. The number of coefficients, in turn, depends on the efficacy of the underlying basis functions, often called dictionaries, in modeling the BRDF. Most, if not all, existing methods either sacrifice the model accuracy to achieve fast reconstruction for real-time applications, or aim for high image fidelity leading to increased storage and computational requirements. Another important aspect is the complexity of the basis functions used in the representation. At one end of the spectrum, we have analytical basis functions such as spherical harmonics and wavelets [Claustres et al. 2003; Ramamoorthi and Hanrahan 2001], which do not require the storage of the basis functions, at the cost of low approximation accuracy, unless a large number of coefficients is used. On the other end of the spectrum, we have decomposition-based methods [Bilgili et al. 2011] that model the BRDF as a multiplication of a set of coefficients and a basis matrix/tensor computed from data. Unfortunately, these approaches require a computationally expensive decomposition, e.g., **Principal Component Analysis (PCA)** or **Singular Value Decomposition (SVD)**, for each BRDF individually and suffer from a high storage cost for the basis itself. Another problem is that the expressiveness of existing bases/decomposition methods is limited. Except for a few, they are in most cases not designed for BRDF data, hence requiring high numbers of coefficients for accurate BRDF representation.

The goal in this article is to develop a new data-driven BRDF model using a set of trained basis functions that enable highly accurate representations with a minimal number of coefficients to represent any BRDF. To solve this challenge, we derive a model that in essence relies on decomposing BRDFs into a coefficient-basis pair, but uses machine learning to adapt the basis to the space of BRDFs to provide maximally sparse coefficients. Sparse BRDF modeling is achieved using a novel BRDF dictionary ensemble and a novel model selection algorithm to efficiently represent a wide range of real-world materials. The learned BRDF dictionary ensemble consists of a set of basis functions trained such that they guarantee a very sparse BRDF representation and near optimal signal reconstruction. Moreover, our model takes into account the multidimensional structure of measured BRDFs (e.g., 3D or 4D depending on the parameterization) and can exploit the information redundancy in the entire BRDF space to reduce the number of coefficients. While in this article, we focus on isotropic materials, our method can be readily applied to anisotropic materials, given a data set of such materials.

The learned ensemble is versatile and can be trained only once to be reused for representing a wide range of previously unseen materials. Additionally, the dictionary ensemble is not restricted to a single BRDF transformation as previous models. Instead, multiple BRDF transformations can be included in the ensemble training such that for each individual BRDF the best representation can be automatically selected and used. This is achieved by a novel model selection method to pick a dictionary in the ensemble that leads to the sparsest solution, the smallest reconstruction error, and the most suitable transformation with respect to rendering

quality. For the experiments and evaluations presented here, we use the MERL [Matusik et al. 2003] and RGL-EPFL [Dupuy and Jakob 2018] databases, which are divided into a training set and a test set used for evaluation.

Another key contribution of this article is a novel algorithm for interpolating two or more BRDFs in the sparse coefficient space. We propose to perform this task in the sparse coefficient space. The key challenge is when the two BRDFs to be interpolated do not lie in the same subspace, i.e., they use different dictionaries of the ensemble. We propose a novel formulation for interpolation of BRDFs that lie in different coordinate systems. The proposed interpolation algorithm admits the construction of a smooth surface over the coefficient space of all BRDFs represented using our model.

The main contributions of this article are as follows:

- A novel non-parametric BRDF model using sparse representations that significantly outperforms existing decomposition-based methods with respect to both model error and rendering quality.
- A multidimensional dictionary ensemble learning method tailored to measured BRDFs.
- A novel BRDF model selection method that chooses the best dictionary for efficient BRDF modeling, as well as the most suitable BRDF normalization function. This enables a unified non-parametric BRDF model regardless of the characteristics of the material.
- A novel algorithm for BRDF interpolation in the sparse coefficient space rather than the BRDF space. We show that two or more measured BRDFs can be linearly interpolated even if each BRDF uses a distinct dictionary in the ensemble.

We compare the proposed non-parametric BRDF model to the current state-of-the-art methods and demonstrate that it performs significantly better in terms of rendering **Signal-to-Noise Ratio (SNR)** and visual quality, as well as having a lower reconstruction error. To the authors' knowledge this is the first BRDF model based on sparse representations and dictionary learning.

Notations. Throughout the article, we use the following notational convention. Vectors and matrices are denoted by boldface lower-case (**a**) and bold-face upper-case letters (**A**), respectively. Tensors are denoted by calligraphic letters, e.g., \mathcal{A} . A finite set of objects is indexed by superscripts, e.g., $\{A^{(i)}\}_{i=1}^N$, whereas individual elements of **a**, **A**, and \mathcal{A} are denoted a_i , A_{i_1, i_2} , $\mathcal{A}_{i_1, \dots, i_n}$, respectively. The determinant of **A** is denoted $\det(\mathbf{A})$. The ℓ_p norm of a vector **s**, for $1 \leq p \leq \infty$, is denoted $\|\mathbf{s}\|_p$. Frobenius norm is denoted $\|\mathbf{s}\|_F$. The ℓ_0 pseudo-norm of a vector, denoted $\|\mathbf{s}\|_0$, defines the number of non-zero elements in the vector. The n -mode product of a tensor **S** and a matrix **U** is denoted $\mathbf{S} \times_n \mathbf{U}$.

2 BACKGROUND AND RELATED WORK

Measured BRDFs have proven to be an important tool in achieving photo-realism during rendering [Dong et al. 2016; Dupuy and Jakob 2018; Matusik et al. 2003]. Even highly complex surfaces such as layered materials require multiple components of measured data to construct novel complex materials [Jakob et al. 2014]. Measured materials, however, are high-dimensional signals with large memory footprint and a key challenge is that small approximation errors can lead to visual artifacts during rendering. To

efficiently represent such high-dimensional measured BRDF data, one can use parametric models, or data-driven models, since densely sampled BRDF data imposes a large memory footprint, making it impractical to use in many applications.

Parametric models. By careful modeling, BRDFs can be encoded with only a few parameters. The components or factors of such models are based on either assumptions describing by the physics of light–surface interactions using, e.g., microfacet theory [Cook and Torrance 1982; Holzschuch and Pacanowski 2017; Walter et al. 2007], or empirical observations of BRDF behaviors [Ashikhmin and Shirley 2000; Blinn 1977; Löw et al. 2012; Nishino and Lombardi 2011; Ward 1992]. However, in many practical cases and applications, existing parametric models cannot accurately fit measured real-world data, as mentioned in Bagher et al. [2016].

Data-driven models. Due to their non-parametric property, data-driven models are superior to parametric models in that the number of degrees of freedom, or implicit model parameters, is much higher. This means that the representative power is higher and the expected approximation error is lower. Factored BRDF models use decomposition techniques to factorize BRDF into several components. Matrix and tensor decompositions have been used by Lawrence et al. [2004], Soler et al. [2018], Bilgili et al. [2011], and Tongbuasirilai et al. [2019]. Moreover, factorization-based models have been used for interactive BRDF editing [Ben-Artzi et al. 2006; Kautz and McCool 1999] and real-time rendering [Soler et al. 2015]. Analytical basis functions have been used for BRDF measurements using spherical harmonics [Ghosh et al. 2009, 2010; Tunwattanapong et al. 2013] and BRDF modeling using rational functions [Pacanowski et al. 2012].

A problem with existing factored models is that rank-1 approximations in most cases lead to inferior results. Accurate modeling requires iterative solutions with many layered factors. Analytic-data-driven BRDF models [Bagher et al. 2016; Sun et al. 2018] employ analytical models extended to a higher number of parameters fitted with measured data to achieve higher accuracy. Cooper et al. [2021] employ the Gaussian mixture model for BRDF representation and clustering that is utilized in the inverse rendering problem. The recent advancement of machine learning algorithms, in particular deep learning, have led to new research directions on BRDF-related topics [Dong 2019]. Deep learning has been used for BRDF editing [Hu et al. 2020], BRDF acquisition [Deschaintre et al. 2018, 2019; Li et al. 2018], compact BRDF representation for importance sampling [Sztrajman et al. 2021], and inverse rendering using invertible neural BRDFs [Chen et al. 2021]. Moreover, deep learning has also been used for **Bidirectional Texture Function (BTF)** compression, see, e.g., Rainer et al. [2019].

Dictionary Learning. One of the most commonly used dictionary learning methods is K-SVD [Aharon et al. 2006], and its many variants [Marsousi et al. 2014; Mazhar and Gader 2008; Mukherjee et al. 2015; Rusu and Dumitrescu 2012], where a 1D signal (i.e., a vector) is represented as a linear combination of a set of basis vectors, called atoms. A clear disadvantage of K-SVD for BRDF representation is signal dimensionality. For instance, a measured BRDF in the MERL data set, excluding the spectral information, is a $90 \times 90 \times 180 = 1,458,000$ -dimensional vector. In practice, the number of data points needed for K-SVD dictionary training should be a multitude of the signal dimensionality to achieve a high-quality

dictionary. In addition to unfeasible computational power required for training, the limited number of available measured BRDF data sets renders the utilization of K-SVD impractical.

In contrast to 1D dictionary learning methods, multidimensional dictionary learning has received only little attention in the literature [Ding et al. 2017; Hawe et al. 2013; Roemer et al. 2014]. In multidimensional dictionary learning, a data point is treated as a tensor, and a dictionary is trained along each mode. For instance, given our example above, instead of training one 1,458,000-dimensional dictionary for the MERL data set, one can train three dictionaries (i.e., one for each mode), where the atom size for these dictionaries are 90, 90, and 180, corresponding to the dimensionality of each mode. To the best of our knowledge, there exists only a few multidimensional dictionary learning algorithms. Our sparse BRDF model in this article is inspired by the multidimensional dictionary ensemble training proposed in Miandji et al. [2019], which has been shown to perform well for high dimensional signals such as light fields and light field videos. We will elaborate on our training scheme for BRDFs in Section 3.2.

3 SPARSE DATA DRIVEN BRDF MODEL

Our non-parametric model is based on learning a set of multidimensional dictionaries, a dictionary ensemble, spanning the space of BRDFs, i.e., the space in which each BRDF is a single multidimensional point. Each dictionary in the ensemble consists of a set of basis functions (i.e., orthonormal matrices), representing each dimension of the BRDF space, that admits sparse representation of any measured BRDF using only a small number of coefficients as illustrated in Figure 1. The dictionary ensemble is trained only once on a given training set of measured BRDFs and can then be reused to represent a wide range of different BRDFs. This is in contrast to previous models that use tensor or matrix decomposition techniques, where the basis and the coefficients are calculated for each BRDF individually.

A major challenge when using machine learning methods, and in particular dictionary learning, on BRDFs is the high dynamic range inherent to the data. In Section 3.1, we describe two data transformations that when applied on measured BRDFs, they improve the fitting to our non-parametric model, see Section 4. The training of the multidimensional dictionaries is described in Sections 3.2 and 3.3, followed by our model selection technique in Section 3.4, where we describe a method to select the most suitable dictionary in the ensemble for any unseen BRDF such that the coefficients are maximally sparse, the modeling error is minimal, and that the data transformation used is one that leads to a better rendering quality. Finally, in Section 3.5, we present a novel algorithm for interpolating two or more measured BRDFs, represented using our model, *directly* in the representation space (i.e., using the sparse coefficients).

A BRDF can be parameterized in many different ways [Barla et al. 2015; Löw et al. 2012; Rusinkiewicz 1998; Stark et al. 2005]. Our dictionary learning approach does not rely on the parameterization of given BRDFs as long as the resolution of these BRDFs is the same. For simplicity and to facilitate a fair comparison to previous works, all the data sets we use here are based on the Rusinkiewicz’s parameterization [Rusinkiewicz 1998] defined by

Table 1. SNR of Rendered Images Using the BRDF Dictionaries Trained with Different Dictionary Sparsity Levels: 32, 64, 128, and 256

Material	Ensemble with $\tau_l = 32$		Ensemble with $\tau_l = 64$		Ensemble with $\tau_l = 128$		Ensemble with $\tau_l = 256$	
	ρ_{t1} SNR(dB)	ρ_{t2} SNR(dB)	ρ_{t1} SNR(dB)	ρ_{t2} SNR(dB)	ρ_{t1} SNR(dB)	ρ_{t2} SNR(dB)	ρ_{t1} SNR(dB)	ρ_{t2} SNR(dB)
blue-fabric	53.9003	58.4393	57.0890	61.1925	56.7419	62.4704	58.7229	62.9932
blue-metallic-paint	<u>54.8105</u>	56.8017	52.4779	59.7930	54.2249	61.0643	52.5073	60.5738
dark-red-paint	44.1094	51.9677	45.8218	52.4098	<u>48.4695</u>	54.7743	46.3005	54.4020
gold-metallic-paint2	<u>46.9514</u>	38.6907	45.6783	36.0324	46.1956	37.4564	42.4208	41.1227
green-metallic-paint2	50.7108	41.8161	49.4635	39.3023	<u>52.8230</u>	43.0459	49.6811	50.2204
light-red-paint	41.4139	49.0550	43.7449	48.7451	<u>47.7306</u>	52.1905	45.1613	50.6002
pink-fabric2	44.8244	49.3862	48.5446	52.5484	<u>52.6230</u>	53.5701	52.5405	54.4938
purple-paint	43.8932	38.8859	42.2491	47.5648	<u>48.2735</u>	48.7324	45.3798	47.1568
red-fabric	47.5606	52.3038	50.9287	54.7831	53.9668	56.7085	<u>55.3687</u>	58.5863
red-metallic-paint	47.2351	40.3386	46.9943	38.4251	<u>49.1860</u>	42.1207	48.6971	42.6229
silver-metallic-paint2	40.3291	42.9256	44.0442	43.2292	44.0323	46.8208	<u>46.4961</u>	44.1504
specular-green-phenolic	48.4841	41.6432	47.3226	36.5157	49.4785	48.8586	49.2522	45.9519
specular-violet-phenolic	48.2384	42.7994	47.4994	37.9801	47.4863	44.5840	<u>48.3638</u>	41.3332
specular-yellow-phenolic	<u>46.4907</u>	39.1758	44.5259	36.1666	45.4146	35.4231	43.1846	36.4720
violet-acrylic	<u>48.7179</u>	44.0610	48.7112	38.9536	47.6828	42.0749	48.1322	36.7368
Average	47.1779	45.8860	47.6730	45.5761	49.6219	48.6596	48.8139	48.4944

Each dictionary has two transformations, ρ_{t1} and ρ_{t2} . The test set consists of 15 MERL materials (not included in the training). The bottom row shows the average SNR over the test set. The underlined numbers are best SNR values for ρ_{t1} and the bold numbers are the best SNR values for ρ_{t2} .

$(\theta_h, \theta_d, \phi_d)$, where θ_h, θ_d are elevation angles of the half vector and the difference vector, respectively, ϕ_d is the azimuthal angle of the difference vector. The sample BRDFs we use here are of resolution $90 \times 90 \times 180$. The data sets are sampled with MERL encoding where a square-root function is applied to θ_h ; see Matusik et al. [2003].

3.1 BRDF Data Transformation

Measured BRDFs often exhibit a very high dynamic range, which introduces many difficulties during parameter fitting and optimization. It is therefore necessary to apply an invertible transformation to the BRDF values using, e.g., a log-mapping as suggested by Löw et al. [2012], Nielsen et al. [2015], Sun et al. [2018], and Tongbuasirilai et al. [2019]. In this article, we use two data transformation functions to improve the performance of our model during training and testing.

The first transformation is based on *log-plus* transformation proposed by Löw et al. [2012]:

$$\rho_{t1}(\omega_h, \omega_d) = \log(\rho(\omega_h, \omega_d) + 1), \quad (1)$$

where ρ is the original BRDF value, ρ_{t1} is the transformed BRDF value, and (ω_h, ω_d) are vectors defining the half vector and the difference vector defined by Rusinkiewicz's parameterization, respectively. For the second transformation, we use the *log-relative mapping* proposed by Nielsen et al. [2015]; however, we exclude the denominator. We call this transformation *cosine-weighted-log*:

$$\rho_{t2}(\omega_h, \omega_d) = \log(\rho(\omega_h, \omega_d) * \mathbf{cosMap}(\omega_h, \omega_d) + 1), \quad (2)$$

where \mathbf{cosMap} is a function that maps the inputs (ω_h, ω_d) directions in the MERL encoding to (ω_i, ω_o) in standard spherical coordinates, and calculates the weights as $\max(\cos(\theta_i) * \cos(\theta_o), \epsilon)$, where $\epsilon = 0.001$.

Using the proposed non-parametric model, we have conducted experiments using both transformations, see Table 1. The *log-plus*

transformation in Equation (1) yields better results when compared to the *cosine-weighted-log* transformation in Equation (2) for glossy materials. The *cosine-weighted-log* is in most cases a better choice for near-diffuse BRDFs.

While we use the two most commonly used BRDF transformations, our sparse BRDF model is not limited to the choice of the transformation function. Indeed, given any new such function, the previously trained dictionary ensemble can be directly applied. However, to further improve the model accuracy, one can train a small set of dictionaries given a training set obtained with the new BRDF transformation. We then take the union of the previously and newly trained ensembles of dictionaries. The expansion of the dictionary ensemble is a unique characteristic of our model. We utilize this property in Section 3.3 to combine different sets of dictionaries, each trained with a distinct training sparsity. The same approach can be used here for improving the model accuracy when a new measured BRDF data set, that requires a more sophisticated transformation, is given.

3.2 Multidimensional Dictionary Learning for BRDFs

To build the non-parametric BRDF model, we seek to accurately model the space of BRDFs using basis functions leading to a high degree of sparsity for the coefficients while maintaining the visual fidelity of each BRDF in the training set. To achieve this, the training algorithm needs to take into account the multidimensional nature of BRDF objects, typically 3D or 4D, depending on the parameterization. Let $\{\mathcal{X}^{(i)}\}_{i=1}^{N_l}$ be a set of N_l BRDFs, where $\mathcal{X}^{(i)} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$. Here, we do not assume any specific parameterization and only require that all the BRDFs in $\{\mathcal{X}^{(i)}\}_{i=1}^{N_l}$ have the same resolution. Moreover, as discussed in Section 3.1, we utilize two BRDF transformations, ρ_{t1} and ρ_{t2} . As a result, the training set consists of two versions of each BRDF.

To achieve a sparse three-dimensional representation of $\{\mathcal{X}^{(i)}\}_{i=1}^{N_l}$, we train an ensemble of K three-dimensional

dictionaries, denoted $\{U^{(1,k)}, U^{(2,k)}, U^{(3,k)}\}_{k=1}^K$, such that each BRDF, $X^{(i)}$, can be decomposed as

$$X^{(i)} = S^{(i)} \times_1 U^{(1,k)} \times_2 U^{(2,k)} \times_3 U^{(3,k)}, \quad (3)$$

where $U^{(1,k)} \in \mathbb{R}^{m_1 \times m_1}$, $U^{(2,k)} \in \mathbb{R}^{m_2 \times m_2}$, $U^{(3,k)} \in \mathbb{R}^{m_3 \times m_3}$, and $k \in \{1, \dots, K\}$. The notation \times_n is used to denote the n -mode product between a tensor and a matrix as described in Kolda and Bader [2009]. Moreover, the tensor $S^{(i)} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ contains a set of sparse coefficients, and we require $\|S^{(i)}\|_0 \leq \tau$, where τ is a user-defined sparsity parameter to achieve a certain sparsity in the representation. Recall from Section 1 that the ℓ_0 norm, denoted $\|\cdot\|_0$, counts the number of nonzero elements in a vector, a matrix, or a tensor. It is evident from Equation (3) that each BRDF is represented using *one* dictionary in the ensemble, in this case $\{U^{(1,k)}, U^{(2,k)}, U^{(3,k)}\}$. Moreover, we require that the number of dictionaries to be much smaller than the number BRDFs in the training set, i.e., $K \ll N_l$. As a result, the training algorithm divides the training set into a set of cluster, where each cluster is represented using one dictionary.

The ensemble training is performed by solving the following optimization problem:

$$\min_{U^{(j,k)}, S^{(i,k)}, M_{i,k}} \sum_{i=1}^{N_l} \sum_{k=1}^K M_{i,k} \|X^{(i)} - S^{(i,k)} \times_1 U^{(1,k)} \times_2 U^{(2,k)} \times_3 U^{(3,k)}\|_F^2, \quad (4a)$$

subject to

$$(U^{(j,k)})^T U^{(j,k)} = I, \quad \forall k = 1, \dots, K, \quad \forall j = 1, \dots, 3, \quad (4b)$$

$$\|S^{(i,k)}\|_0 \leq \tau_l, \quad (4c)$$

$$\sum_{k=1}^K M_{i,k} = 1, \quad \forall i = 1, \dots, N_l, \quad (4d)$$

where the matrix $M \in \mathbb{R}^{N_l \times K}$ is a clustering matrix associating each BRDF in the training set to one multidimensional dictionary in the ensemble. Equation (4b) ensures the orthonormality of each multidimensional dictionary, the sparsity of the coefficients is enforced by Equation (4c), and the single dictionary representation is enforced by the constraint in Equation (4c). The user-defined parameter τ_l defines the training sparsity. It should be noted that the clustering matrix M divides the BRDFs in the training set into a set of clusters such that optimal sparse representation is achieved with respect to the number of model parameters (or coefficients) and the representation error. This clustering is an integral part of our model and improves the accuracy of BRDF representations. The supplementary document accompanying this manuscript contains the update rules for solving Equation (4) iteratively with respect to $U^{(j,k)}$, $S^{(i,k)}$, and $M_{i,k}$.

Our sparse BRDF modeling is inspired by the **Aggregate Multidimensional Dictionary Ensemble (AMDE)** proposed by Miandji et al. [2019]. However, we do not perform pre-clustering of data points, in this case BRDFs, for the following two reasons: First, the number of existing measured BRDF data sets is very limited. Hence, if we apply pre-clustering, then the number of available

BRDFs to train a dictionary ensemble becomes inadequate. Second, since we use each BRDF as a data point, the size of each data point is $90 * 90 * 180 = 1,458,000$, hence rendering the proposed pre-clustering method in Miandji et al. [2019] impractical. Indeed, the two BRDF transformations discussed in Section 3.1 can be seen as a pre-clustering of the training set. These transformations divide the training set into diffuse and glossy BRDFs. Moreover, as it will be described in Section 3.3, and unlike the method of Miandji et al. [2019], we perform multiple trainings of the same training set but with different values for the training sparsity parameter, τ_l . The obtained ensembles are combined to form an ensemble that can efficiently represent BRDFs with less reconstruction error.

3.3 BRDF Dictionary Ensemble with Multiple Sparsities

Measured BRDFs exhibit a variable degree of sparsity in the representation space (also known as the coefficient space). Indeed given a suitable dictionary, a diffuse material requires only a small number of coefficients while a highly glossy BRDF needs a significantly higher number of coefficients for an accurate representation. This phenomenon has been observed by previous work on non-parametric modeling of BRDFs based on factorization or using commonly known basis functions such as spherical harmonics [Lawrence et al. 2004; Nielsen et al. 2015; Sun et al. 2018; Tunwatanapong et al. 2013]. A shortcoming of the dictionary ensemble learning method described in Section 3.2 is that we do not take into account the intrinsic sparsity of various materials in the training set. In other words, since the training sparsity τ_l is fixed for all the BRDFs in the training set, a small value for τ_l will steer the optimization algorithm to more efficiently model low frequency (or diffuse-like) materials, while neglecting high frequency materials. Moreover, when τ_l is large, the dictionary optimization adapts to high frequency materials, leading to degradation of quality for diffuse materials due to over-fitting. Indeed, finding the optimal values of τ_l for a set of BRDFs is a tedious task, and requires various assumptions on the behavior of these materials in the representation space.

Instead of finding the optimal values for τ_l for a large and diverse set of BRDFs in the training set, we train multiple ensembles of dictionaries, each with a different value for τ_l , so that we can model both low and high frequency details of the training BRDFs more efficiently, while lowering the risk of over-fitting. After training each ensemble according to the method described in Section 3.2, we combine them all to form one ensemble that includes all the dictionaries. We then utilize our model selection algorithm, see Section 3.4, for selecting the optimal dictionary in the combined ensemble for a given BRDF. Regardless of the material properties, e.g., low or high frequency, our model selection is able to select the best dictionary in the ensemble for the vast majority of cases, see Section 4 for details. In this article, we train 4 ensembles, each with 8 dictionaries, which are trained with $\tau_l = 32$, $\tau_l = 64$, $\tau_l = 128$, and $\tau_l = 256$; hence, the final ensemble consists of 32 dictionaries. Note that the combined ensemble with 32 dictionaries only requires $(32 * (90 * 90 + 90 * 90 + 180 * 180) * 4) / 1,024^2 = 5.93$ MB of storage, assuming 4 bytes for each element. Therefore, our ensemble can be used in applications where storage capabilities are

Table 2. Rendering SNR, Gamma-mapped-MSE, and MSE, Obtained Using Our Sparse BRDF Model for ρ_{t1} and ρ_{t2}

Material	Rendering SNR (dB)		Gamma-mapped-MSE		MSE	
	Our ρ_{t1}	Our ρ_{t2}	Our ρ_{t1}	Our ρ_{t2}	Our ρ_{t1}	Our ρ_{t2}
blue-fabric	53.99	62.16	0.0233	0.0038	0.0002	0.0062
blue-metallic-paint	51.65	60.53	0.0448	0.0375	0.0011	0.0317
dark-red-paint	49.16	54.80	0.0616	0.0242	0.0295	0.1209
gold-metallic-paint2	48.29	37.68	0.9248	0.9350	72.0330	38.7850
green-metallic-paint2	57.48	43.36	0.8767	0.8939	31.5660	11.9140
light-red-paint	46.51	51.68	0.0552	0.0312	0.0567	0.1699
pink-fabric2	52.66	52.71	0.0230	0.0125	0.0003	0.0323
purple-paint	45.81	47.24	0.1991	0.1725	2.8225	2.1342
red-fabric	56.26	55.05	0.0177	0.0078	0.0002	0.0172
red-metallic-paint	52.70	42.62	1.2910	1.3086	45.0140	19.3610
silver-metallic-paint2	44.70	44.55	0.0988	0.0895	0.0029	0.1217
specular-green-phenolic	53.09	36.67	0.9889	1.0161	27.1120	14.0560
specular-violet-phenolic	50.51	38.21	0.9722	0.9925	22.4920	14.34100
specular-yellow-phenolic	46.81	36.40	0.9454	0.9686	18.8780	10.6280
violet-acrylic	50.07	42.61	0.7770	0.7849	20.2560	12.7940

For each error metric, the best result between ρ_{t1} and ρ_{t2} is shown by bold numbers. Comparing the chosen transformation based on rendering SNR with Gamma-mapped-MSE and MSE in the BRDF space, we see that the Gamma-mapped-MSE can well distinguish the suitable transformation for 13 of 15 materials. It can also be seen that MSE only selects the correct transformation for 3 of 15 materials. For results generated using Gamma-mapped-MSE, we set $\gamma = 2.0$.

very limited. In Table 1, we present rendering SNR results obtained from ensembles trained with different values for the training sparsity, τ_l .

Our BRDF model represents each BRDF using one dictionary from the ensemble. Each dictionary can be seen as a set of basis functions spanning the BRDF manifold in a local neighborhood defined by the dictionary. Therefore, our ensemble of 32 dictionaries defines 32 local coordinate frames for representing the space of all BRDFs defined on a manifold. This is in contrast to the PCA dictionary used in Nielsen et al. [2015], where a single coordinate frame defined the BRDF manifold, or the diffuse-specular separation method [Sun et al. 2018], where two coordinate frames representing diffuse and specular properties are defined. Hence, our method, where the number of dictionaries or coordinate frames, is more flexible and expressive to cover a greater range of material classes. Indeed, the membership matrix, \mathbf{M} , can be thought of as a clustering matrix for the BRDF manifold. During training, a BRDF is assigned to a dictionary in which it is best represented given the limited number of coefficients defined by τ_l . The different dictionaries represent different physical properties of BRDFs. Table 1 demonstrates that, e.g., phenolic and paint materials cluster in dictionaries with low sparsity and that more glossy or specular BRDFs reside in high sparsity dictionaries. In addition, our ensemble can be expanded by training more dictionaries and adding them to the existing ensemble. This is particularly useful for modeling new data sets or new data transformations, if the existing ensemble does not provide satisfactory results.

3.4 BRDF Model Selection

Once the ensemble of dictionaries is trained, the next step is to use it for the sparse representation of BRDFs. We call this stage *model selection*, since our dictionaries in the ensemble are trained with different transformations and sparsities, we need to find one dictionary that leads to the most sparse coefficients with the least error, as well as the best performing transformation between ρ_{t1} and ρ_{t2} .

Indeed, as mentioned in Section 3.1, our method is not limited to the number of transformations. In what follows, the term *ensemble* refers to the combined ensemble that is trained with multiple sparsities and BRDF transformations, as described in Section 3.3.

We begin by describing our method for selecting the most suitable dictionary in the ensemble for BRDF representation. This can be achieved by projecting each BRDF onto all the dictionaries in the ensemble. Let $\{\mathcal{Y}^{(i)}\}_{i=1}^{N_t}$ be a BRDF test set. The projection step is formulated as

$$\hat{\mathcal{S}}^{(i,k)} = \mathcal{Y}^{(i)} \times_1 (\mathbf{U}^{(1,k)})^T \times_2 (\mathbf{U}^{(2,k)})^T \times_3 (\mathbf{U}^{(3,k)})^T, \quad (5)$$

where $\mathcal{Y}^{(i)}$ is a BRDF in the testing set that we like to obtain a sparse representation of. As discussed in Section 3.1, we utilize two BRDF transformations, ρ_{t1} and ρ_{t2} . As a result, the testing set consists of two versions of each BRDF $\{\mathcal{Y}^{(i)}\}_{i=1}^{N_t}$. The smallest components in the coefficient tensors $\hat{\mathcal{S}}^{(i,k)}$ are progressively nullified until we reach a user defined sparsity level, called the testing sparsity, τ_t , or when the representation error becomes larger than a user defined threshold. The testing sparsity, which defines the model complexity, is different than the training sparsity τ_l and we typically require $\tau_t \geq \tau_l$. Generally, a higher value for τ_t is required for glossy materials than for diffuse to achieve an accurate BRDF representation. Indeed, the number of coefficients provides a trade-off between quality and performance, making our model flexible enough to be applied in a variety of applications.

After sparsifying $\hat{\mathcal{S}}^{(i,k)}$, $\forall k \in \{1, \dots, K\}$, we pick the dictionary corresponding to the sparsest coefficient tensor among $\hat{\mathcal{S}}^{(i,k)}$, $\forall k \in \{1, \dots, K\}$. If all the coefficient tensors $\hat{\mathcal{S}}^{(i,k)}$, $\forall k \in \{1, \dots, K\}$ achieve the same sparsity, then we pick the dictionary corresponding to the least reconstruction error. Note that since the number of BRDFs in a test set, which contains the set of all possible BRDFs with the same resolution, is always larger than the number of dictionaries in the ensemble, i.e., $N_t > K$, the dictionary selection

algorithm described above clusters the test set based on the dictionaries such that each cluster of BRDFs in the test set uses one dictionary of the ensemble. Recall that the same property holds for the training set since $N_l \gg K$.

The reconstruction error for a BRDF in the test set, $\mathbf{y}^{(i)}$, modeled using a dictionary $\{\mathbf{U}^{(1,k)}, \mathbf{U}^{(2,k)}, \mathbf{U}^{(3,k)}\}$, $k \in \{1, \dots, K\}$, is simply calculated as

$$\left\| \mathbf{y}^{(i)} - \hat{\mathbf{S}}^{(i,k)} \times_1 \mathbf{U}^{(1,k)} \times_2 \mathbf{U}^{(2,k)} \times_3 \mathbf{U}^{(3,k)} \right\|_F^2. \quad (6)$$

Because the BRDF dictionary ensemble is trained once and can be used for the sparse representation of unobserved BRDFs, the storage cost of the model in Equation (3) is defined by the storage complexity of the sparse coefficient tensor $\hat{\mathbf{S}}^{(i)}$ in Equation (5). We store the nonzero elements in $\hat{\mathbf{S}}^{(i)}$ as tuples of nonzero element locations and values, denoted $\{l_t^1, l_t^2, l_t^3, \hat{\mathbf{S}}_{l_t^1, l_t^2, l_t^3}^{(i)}\}_{t=1}^{\tau_t}$, where the indices l_t^1, l_t^2 , and l_t^3 store the location of the t th nonzero element of $\hat{\mathbf{S}}^{(i)}$, while the corresponding value is $\hat{\mathbf{S}}_{l_t^1, l_t^2, l_t^3}^{(i)}$.

The reconstruction of a given BRDF, $\mathbf{y}^{(i)}$, using our model is computed by multiplying the sparse coefficient tensor $\hat{\mathbf{S}}^{(i,k)}$ with the corresponding dictionary as follows:

$$\hat{\mathbf{y}}^{(i)} = \hat{\mathbf{S}}^{(i,k)} \times_1 \mathbf{U}^{(1,k)} \times_2 \mathbf{U}^{(2,k)} \times_3 \mathbf{U}^{(3,k)}, \quad (7)$$

where k is the index of the dictionary chosen by the model selection method described above. Thanks to the fact that the coefficient tensor $\hat{\mathbf{S}}^{(i,k)}$ is sparse, Equation (7) is computationally tractable even for real-time applications. Indeed, we can evaluate Equation (7) by only multiplying the nonzero elements of $\hat{\mathbf{S}}^{(i,k)}$ with the dictionary, as shown in Miandji et al. [2019]. Reconstructing a single element in a BRDF only requires $3\tau_t$ operations of type **multiply-add (MAD)**. Indeed, for ray tracing applications, we only to reconstruct one or a few elements of the BRDF tensor at each ray intersection point. Moreover, since τ_t is user-defined, our method is computationally flexible and can provide an intuitive that makes it suitable for a variety of applications.

Since our dictionary is trained with two sets of transformed BRDFs, i.e., ρ_{t1} and ρ_{t2} , we can obtain two reconstructed BRDFs from an unseen BRDF by employing the algorithm described above. This still leaves us with the problem of selecting the best reconstructed BRDF between ρ_{t1} and ρ_{t2} . Due to the discrepancy between quantitative quality metrics computed over the BRDF space (such as MSE) and the rendering quality [Bieron and Peers 2020], model selection is a difficult task for BRDF fitting, as well as learning-based methods such as ours. For instance, log-based metrics [Löw et al. 2012; Sun et al. 2018] have been used to improve efficiency of fitting measured BRDFs to parametric functions. Indeed, the most reliable technique is to render a collection of images for all possible variations of the model and select one that is closest to an image rendered using the reference BRDF. This approach has been used by Bieron et al. [2020] for BRDF fitting. To reduce the number of renderings, multiple BRDF parameter fitting are performed using a power function with different inputs. The model selection is then performed by rendering a test scene and choosing the best model based on image quality metrics.

We propose a model selection approach that does not require rendering the reconstructed ρ_{t1} and ρ_{t2} variants of BRDFs. From our observations, we found that the MSE as an error metric in the model selection algorithm, see Equation (6), does not match a selection method based on rendering quality to choose from ρ_{t1} and ρ_{t2} , see Table 2. To address this problem, we use a Gamma mapping function, $\Gamma(\rho, \gamma) = \rho^{1/\gamma}$, on the reference ρ_{t1} , and ρ_{t2} , prior to computing the MSE. We refer to this error metric as *Gamma-mapped-MSE*. Note that since the reference BRDF is in linear BRDF domain, i.e., it is not transformed, we invert ρ_{t1} and ρ_{t2} according to Equations (1) and (2), respectively, prior to computing the Gamma-mapped-MSE.

In Table 2, we report reconstruction quality measured with rendering SNR, Gamma-mapped-MSE, and MSE for both ρ_{t1} and ρ_{t2} . For these results, we used 15 test materials from the MERL data set, while the remaining 85 materials were used for training. For each error metric, the best result is highlighted in bold-face characters. It can be seen that Gamma-mapped-MSE can well distinguish the best transformation among ρ_{t1} and ρ_{t2} with respect to rendering SNR for 13 of 15 materials. The two exceptions are *red-fabric* and *silver-metallic-paint2*. It can also be seen that MSE only selects the correct transformation for 3 of 15 materials. To obtain Gamma-mapped-MSE results, we used $\gamma = 2.0$. Indeed, this parameter can be tuned per-BRDF to further improve our results; however, we found that fixed value of $\gamma = 2.0$ is adequate to achieve a significant advantage over previous methods. Compared to Bieron and Peers [2020], our model selection approach provides greater flexibility by considering rendering quality (via *Gamma-mapped-MSE*), BRDF transformations (using ρ_{t1} and ρ_{t2}), as well as model complexity (via τ_t).

3.5 BRDF Interpolation in Sparseland

In this section, we describe our method for interpolating two or more BRDFs, where we consider two scenarios: when the BRDFs use the same dictionary from the ensemble, and when each BRDF uses a distinct dictionary. Unlike previous methods, we perform the interpolation in the lower-dimensional space defined by the sparse coefficients, as opposed to interpolating the BRDFs in the original high-dimensional BRDF-space. We show that using our model, one can efficiently interpolate between BRDFs even when they use different dictionaries. BRDF interpolation on a nonlinear space has been previously explored using the charting algorithm [Brand 2002; Matusik et al. 2003], where locally linear lower-dimensional subspaces are constructed with the same dimensionality. Here, we show that when two or more BRDFs are sparse in distinct dictionaries, BRDF interpolation in the sparse coefficient space is possible. Moreover, each BRDF that is used for interpolation using our model can have different sparsity. Compared to Brand [2002] and Matusik et al. [2003], the local subspaces in our model can have different dimensionalities in the representation space. In what follows, and without loss of generality, we present our interpolation method for two BRDFs. Extension of our formulation to three or more BRDF data points is straightforward.

As described in Sections 1 and 2, existing non-parametric models obtain the model coefficients by fitting techniques that are applied for each BRDF individually. Therefore, the resulting

coefficients for each BRDF do not necessarily share a common basis. Since the coefficients of each BRDF to be interpolated lie in a distinct coordinate system, one cannot perform the interpolation in the coefficient space. It should be noted that, once a BRDF is represented using a non-parametric model, we do not have access to the original BRDF. The only information that is available is the coefficients and the representation basis (i.e., the dictionary). Therefore, it is essential to perform the interpolation in the coefficient space.

To interpolate non-parametric BRDFs in the coefficient space, one should define a common basis for the BRDFs to be interpolated, e.g., by employing PCA as in Nielsen et al. [2015]. Since all the BRDF lie in the same coordinate space, defined by the principal components, one can linearly interpolate the coefficients instead of the original BRDFs. Note that this property is not shared by the majority of non-parametric models [Bagher et al. 2016; Bilgili et al. 2011]. The interpolated coefficients and the principal components are then used to reconstruct the resulting BRDF. However, as it will be shown in Section 4, having one common basis (i.e., the set of principal components) for the entire space of possible BRDFs leads to vastly inferior results due to inadequate model capacity. If we define multiple dictionaries, as with our proposed method, then two BRDFs that are represented with two distinct dictionaries cannot be interpolated in the coefficient space. This is evident, since these two BRDFs use two distinct coordinate systems. Our proposed interpolation technique overcomes this limitation by transforming the sparse coefficients into a common basis, which enables the interpolation directly in the coefficient space.

Let $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ be three BRDFs that, respectively, use the dictionaries $\{\mathbf{U}^{(1,1)}, \mathbf{U}^{(2,1)}, \mathbf{U}^{(3,1)}\}$, $\{\mathbf{U}^{(1,1)}, \mathbf{U}^{(2,1)}, \mathbf{U}^{(3,1)}\}$, and $\{\mathbf{U}^{(1,2)}, \mathbf{U}^{(2,2)}, \mathbf{U}^{(3,2)}\}$, respectively; i.e., we assume that \mathcal{B}_1 and \mathcal{B}_2 use the same dictionary from the ensemble. The linearly interpolated BRDF obtained from \mathcal{B}_1 and \mathcal{B}_2 , which we denote \mathcal{I} , is then given by

$$\mathcal{I} = (1 - \alpha)\mathcal{B}_1 + \alpha\mathcal{B}_2 \quad (8)$$

$$= (1 - \alpha)\mathcal{S}_1 \times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)} \\ + \alpha\mathcal{S}_2 \times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)} \quad (9)$$

$$= ((1 - \alpha)\mathcal{S}_1 + \alpha\mathcal{S}_2) \times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)}, \quad (10)$$

where \mathcal{S}_1 and \mathcal{S}_2 are the sparse coefficients of \mathcal{B}_1 and \mathcal{B}_2 , respectively; the parameter α is the interpolation coefficient. Note that, Equation (10) follows from Equation (9) only because the dictionary $\{\mathbf{U}^{(1,1)}, \mathbf{U}^{(2,1)}, \mathbf{U}^{(3,1)}\}$ is used by both \mathcal{B}_1 and \mathcal{B}_2 . One can apply the same approach for bilinear and bicubic interpolation, or even interpolation of BRDFs over triangles using barycentric coordinates.

To interpolate two BRDFs that do not share the same basis, i.e., when they are modeled using two distinct dictionaries from the ensemble, one cannot utilize Equation (10). For this case, we propose to transform the sparse coefficients of all BRDF data points that are used in the interpolation into a common basis by choosing one dictionary as the reference basis. In this way, we can utilize Equation (10) for interpolation in the coefficient space rather than the BRDF space. To elaborate on this, and without loss of generality, let us present our approach for interpolating \mathcal{B}_1 and

\mathcal{B}_3 , which use two distinct dictionaries $\{\mathbf{U}^{(1,1)}, \mathbf{U}^{(2,1)}, \mathbf{U}^{(3,1)}\}$ and $\{\mathbf{U}^{(1,3)}, \mathbf{U}^{(2,3)}, \mathbf{U}^{(3,3)}\}$, respectively. Our method is based on a simple observation: Given two orthonormal matrices, say \mathbf{D}_1 and \mathbf{D}_2 , of the same dimensionality, there exists a unique orthonormal transformation matrix, \mathbf{R} , such that $\mathbf{D}_1 = \mathbf{D}_2\mathbf{R}$ or $\mathbf{D}_2 = \mathbf{D}_1\mathbf{R}^T$. More concretely,

PROPOSITION 1. *Let \mathbf{A} and \mathbf{B} be two distinct orthonormal matrices. Then the solution of*

$$\min_{\mathbf{R}} \|\mathbf{B} - \mathbf{A}\mathbf{R}\|_F^2 \quad \text{subject to} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}, \quad (11)$$

is $\mathbf{R} = \mathbf{A}^T\mathbf{B}$.

The proof of Proposition 1 can be found in Gower and Dijksterhuis [2004]. Note that Proposition 1 is a variant of the Orthogonal Procrustes problem with the difference that here we assume that the matrices \mathbf{A} and \mathbf{B} are orthonormal; as a result, the transformation matrix \mathbf{R} is either a rotation or rotation-reflection matrix, depending on whether $\det(\mathbf{R}) = 1$ or $\det(\mathbf{R}) = -1$.

Let $\mathbf{R}^{(i,k) \rightarrow (i,j)}$ be the transformation matrix that transforms the orthonormal matrix $\mathbf{U}^{(i,k)}$ to $\mathbf{U}^{(i,j)}$; i.e., we have that $\mathbf{U}^{(i,j)} = \mathbf{U}^{(i,k)}\mathbf{R}^{(i,k) \rightarrow (i,j)}$. It should be noted that, according to Proposition 1, we can only compute $\mathbf{R}^{(i,k) \rightarrow (i,j)}$ for two orthonormal matrices of the same dimensionality. This is reflected in the fact that the index i is shared by $\mathbf{U}^{(i,k)}$ and $\mathbf{U}^{(i,j)}$, i.e., the matrices that define an orthonormal basis for the i th mode of BRDFs. The transformation matrices, to be obtained using Proposition 1, between the dictionaries for \mathcal{B}_1 and \mathcal{B}_3 are

$$\mathbf{U}^{(1,2)} = \mathbf{U}^{(1,1)}\mathbf{R}^{(1,1) \rightarrow (1,2)}, \quad (12)$$

$$\mathbf{U}^{(2,2)} = \mathbf{U}^{(2,1)}\mathbf{R}^{(2,1) \rightarrow (2,2)}, \quad (13)$$

$$\mathbf{U}^{(3,2)} = \mathbf{U}^{(3,1)}\mathbf{R}^{(3,1) \rightarrow (3,2)}. \quad (14)$$

We can now state the formula for interpolating between two BRDFs that are not represented by a common dictionary, namely, \mathcal{B}_1 and \mathcal{B}_3 as given by the example above:

$$\mathcal{I} = (1 - \alpha)\mathcal{B}_1 + \alpha\mathcal{B}_3 \quad (15)$$

$$= (1 - \alpha)\mathcal{S}_1 \times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)} \\ + \alpha\mathcal{S}_3 \times_1 \mathbf{U}^{(1,2)} \times_2 \mathbf{U}^{(2,2)} \times_3 \mathbf{U}^{(3,2)} \quad (16)$$

$$= (1 - \alpha)\mathcal{S}_1 \times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)} \\ + \alpha\mathcal{S}_3 \times_1 \mathbf{U}^{(1,1)}\mathbf{R}^{(1,1) \rightarrow (1,2)} \\ \times_2 \mathbf{U}^{(2,1)}\mathbf{R}^{(2,1) \rightarrow (2,2)} \times_3 \mathbf{U}^{(3,1)}\mathbf{R}^{(3,1) \rightarrow (3,2)} \quad (17)$$

$$= (1 - \alpha)\mathcal{S}_1 \times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)} \\ + \alpha \underbrace{(\mathcal{S}_3 \times_1 \mathbf{R}^{(1,1) \rightarrow (1,2)} \times_2 \mathbf{R}^{(2,1) \rightarrow (2,2)} \times_3 \mathbf{R}^{(3,1) \rightarrow (3,2)})}_{\tilde{\mathcal{S}}_3} \quad (18)$$

$$\times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)}, \quad (18)$$

$$= ((1 - \alpha)\mathcal{S}_1 + \alpha\tilde{\mathcal{S}}_3) \times_1 \mathbf{U}^{(1,1)} \times_2 \mathbf{U}^{(2,1)} \times_3 \mathbf{U}^{(3,1)}, \quad (19)$$

where Equation (18) follows from Equation (17) due to a fundamental property of the n-mode product [Kolda and Bader 2009]. Equation (18) shows that one can apply the transformation matrices onto the coefficients of \mathcal{B}_3 to transform them into the coordinate space of \mathcal{B}_1 defined by $\{\mathbf{U}^{(1,1)}, \mathbf{U}^{(2,1)}, \mathbf{U}^{(3,1)}\}$. When we have

more than two BRDFs to interpolate, where each BRDF uses a distinct dictionary from the ensemble, we choose one dictionary as the reference basis and utilize Equation (18) to transform all coefficients to this reference dictionary. Afterwards, linear, bilinear or cubic interpolation can be done on coefficients directly. The results are then multiplied by the reference dictionary to obtain the interpolated BRDF, as in Equation (19).

It can be noted from Equation (19) that $\tilde{\mathcal{S}}_3$ is not necessarily sparse, since it is obtained from multiplying the sparse tensor \mathcal{S}_3 by the dense matrices $\{\mathbf{R}^{(1,1) \rightarrow (1,2)}, \mathbf{R}^{(2,1) \rightarrow (2,2)}, \mathbf{R}^{(3,1) \rightarrow (3,2)}\}$. To achieve a desired sparsity for the interpolated BRDF, we utilize the projection step of the model selection algorithm in Section 3.4. Note that since the dictionary is known, i.e., $\{\mathbf{U}^{(1,1)}, \mathbf{U}^{(2,1)}, \mathbf{U}^{(3,1)}\}$, we do not need to perform the full model selection algorithm. This operation can be thought of as re-projecting the interpolated BRDF onto the sparse BRDF manifold. Moreover, it allows us to obtain a desired sparsity for the interpolated BRDF, since the re-projection sparsity can be different than the sparsity of the BRDFs used for interpolation. It should be noted that our interpolation algorithm does not require equal sparsity for the BRDFs used in the interpolation.

The interpolation algorithm proposed above admits the construction of a smooth surface over the coefficient space of all BRDFs represented using our model. Since our model enables the efficient sparse representation of any measured BRDF, any two (or more) BRDFs can be smoothly interpolated in the coefficient space using our model.

3.6 Importance Sampling

For the results presented in this article, we utilize the Rusinkiewicz's parameterization defined by $(\theta_h, \theta_d, \phi_d)$. As a result, **Multiple Importance Sampling (MIS)** can be utilized in the same way as the method presented in Bagher et al. [2016]. A two-dimensional **Cumulative Distribution Function (CDF)** can be constructed from our sparse BRDF representation using the luminance channel. The inverted CDF is then used to sample directions proportional to the BRDF. For the example renderings in Section 4 where HDR environment maps are used, this is also combined with a two-dimensional CDF for sampling the illumination from the environment maps. The multiple importance sampling strategies are combined using MIS as suggested in Bagher et al. [2016].

4 RESULTS AND DISCUSSION

This section presents an evaluation of the proposed BRDF model and comparisons to the current state-of-the-art models in terms of BRDF reconstruction error and rendering quality. The rendering results were generated using PBRT [Pharr and Humphreys 2010] with the Grace Cathedral environment map. The images were rendered at a resolution of 512×512 pixels using 512 pixel samples in PBRT with the *directlighting* surface integrator and 256 *infinite* light-source samples.

The BRDF dictionary was trained using materials from the MERL database [Matusik et al. 2003] and RGL-EPFL isotropic BRDF database [Dupuy and Jakob 2018]. We split the MERL and RGL-EPFL materials into a training set and a test set. The training

set contains 136 materials, where 85 materials are from the MERL dataset and 51 materials are from the RGL-EPFL dataset. The test set contains 28 materials with 15 materials from the MERL dataset, 8 materials from the DTU data set [Nielsen et al. 2015] and 5 materials from RGL-EPFL. The training and test sets cover a wide range of material classes and none of the materials in the test set appear in the training set.

Each BRDF color channel is processed independently for the training and model selection. We use the Rusinkiewicz's parameterization [Rusinkiewicz 1998], at a resolution of $90 \times 90 \times 180$, i.e., we have $m_1 = 90$, $m_2 = 90$, and $m_3 = 180$. For our experiments, we trained four ensembles, each with $K = 8$ dictionaries and with training sparsities of $\tau_l = 32$, $\tau_l = 64$, $\tau_l = 128$, and $\tau_l = 256$. We then construct one ensemble by taking the union of the dictionaries in the four ensembles that were trained, as described in Section 3.3. The training BRDFs were transformed using *log-plus* (ρ_{t1}) and *cosine-weighted-log* (ρ_{t2}) functions before starting the training, hence resulting in 272 materials. The training took about 11 hours on a machine with 40 cores clocked at 2.4 GHz.

Once the ensemble is trained, we use the model selection algorithm, described in Section 3.4, to obtain the optimal sparse coefficients of each BRDF in the test set. The model selection for each BRDF and for all three color channels takes about 61 seconds using a CPU with 16 cores clocked at 3.8 GHz. Note that the model selection can be parallelized over the BRDFs in the test set, since this task is performed independently for each BRDF. Moreover, the reconstruction of all the elements of a measured BRDF, i.e., a total of 4,374,000 elements, takes 17 seconds using the same hardware setup. It should be noted that during rendering, only one or a few values from the measured BRDF is needed at each intersection point. In this case, we utilize the algorithm proposed in Mianji et al. [2019] to recover a single element. In addition, since the reconstruction in Equation (7) is based on the n -mode product of a sparse tensor and dense matrices, we can utilize the GPU-based algorithm in Baravdish et al. [2019] for real-time applications. We have left this application for future work. A CPU-based implementation of our method for model selection and reconstruction is provided at <https://github.com/emianji/sparse-brdf>.

To evaluate our sparse BRDF model, we use two quality metrics: SNR that is calculated on the rendered images (floating-point images) and **Relative Absolute Error (RAE)**, which is computed on linear BRDF values. Note that since the reference BRDF is in linear BRDF domain, i.e., it is not transformed, we invert the transformation of the reconstructed BRDF (which may be ρ_{t1} or ρ_{t2} depending on the model selection result) to transform it back to the linear space prior to computing the RAE. The RAE is defined as

$$RAE = \sqrt{\frac{\sum (\rho_{ref} - \rho_{recon})^2}{\sum (\rho_{ref}^2)}}, \quad (20)$$

where ρ_{ref} is the reference BRDF, and ρ_{recon} is the reconstructed BRDF. Even though rendering SNR (or PSNR) is typically used to evaluate BRDF models in many publications, RAE is very useful to capture the model accuracy without relying on a specific rendering setup. This is because RAE is computed on all BRDF values, while rendering SNR only takes into account a portion of the BRDF values, depending on the rendering setup. Note that for

Table 3. Average, Standard Deviation, Minimum, and Maximum Rendering SNR Values of Each BRDF Model Obtained from 15 Materials in the MERL Dataset

BRDF Model	Average SNR (dB)	Standard Deviation	Minimum SNR (dB)	Maximum SNR (dB)
Ours, $\tau_t = 262$, using only ρ_{t1}	50.65	3.8106	44.70	57.49
Ours, $\tau_t = 262$, using only ρ_{t2}	47.08	8.5939	36.40	62.16
Ours, $\tau_t = 262$, using Gamma-mapped-MSE	52.51	4.9752	44.55	62.16
Bagher et al. [2016]	42.76	11.6323	27.11	63.88
Bilgili et al. [2011]	32.63	5.8724	22.86	43.17
Tongbuasirilai et al. [2019] using CPD-PDV rank-1 ($L = 1$)	33.83	5.5236	22.22	42.71
Tongbuasirilai et al. [2019] using CPD-HD rank-1 ($L = 1$)	32.51	8.4995	22.97	52.27

None of these materials were included in our training set. Yet, our method significantly outperforms state-of-the-art decomposition-based methods, such as Bagher et al. [2016], where the basis and coefficients should be computed for each given BRDF (i.e., the training and testing sets are not distinct).

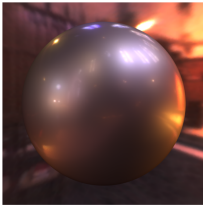

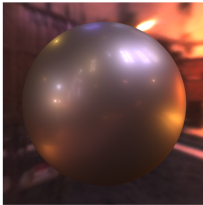
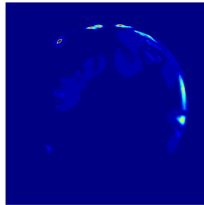
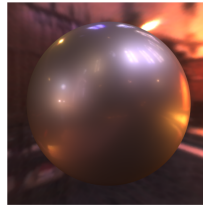

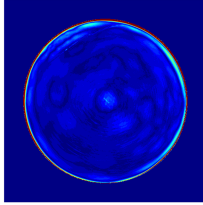

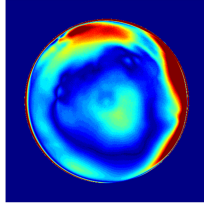

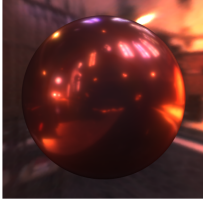

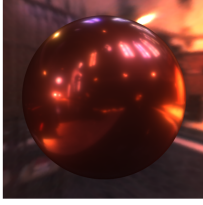
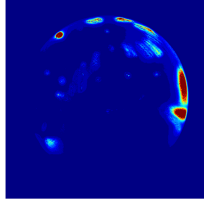
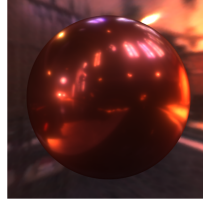
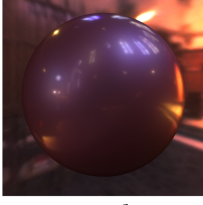


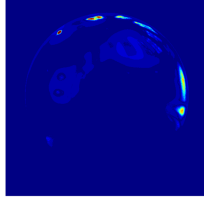
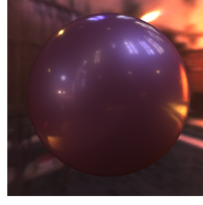
Material	Our Model	absolute error	Bagher et al.	absolute error	reference
gold-metallic-paint2	 48.29dB		 29.77dB		
red-fabric	 55.05dB		 51.44dB		
red-metallic-paint	 52.70dB		 34.06dB		
violet-acrylic	 50.07dB	 0 1	 31.68dB	 0 1	

Fig. 2. Visual quality of selected materials from the MERL test set. For our model, we used $\tau_t = 262$ coefficients to compare with the method of Bagher et al. [2016]. Renderings are gamma corrected for display. The error images have been multiplied by 10.0 for visual comparisons. All images have been rendered using PBRT [Pharr and Humphreys 2010] with the *Grace Cathedral* environment map.

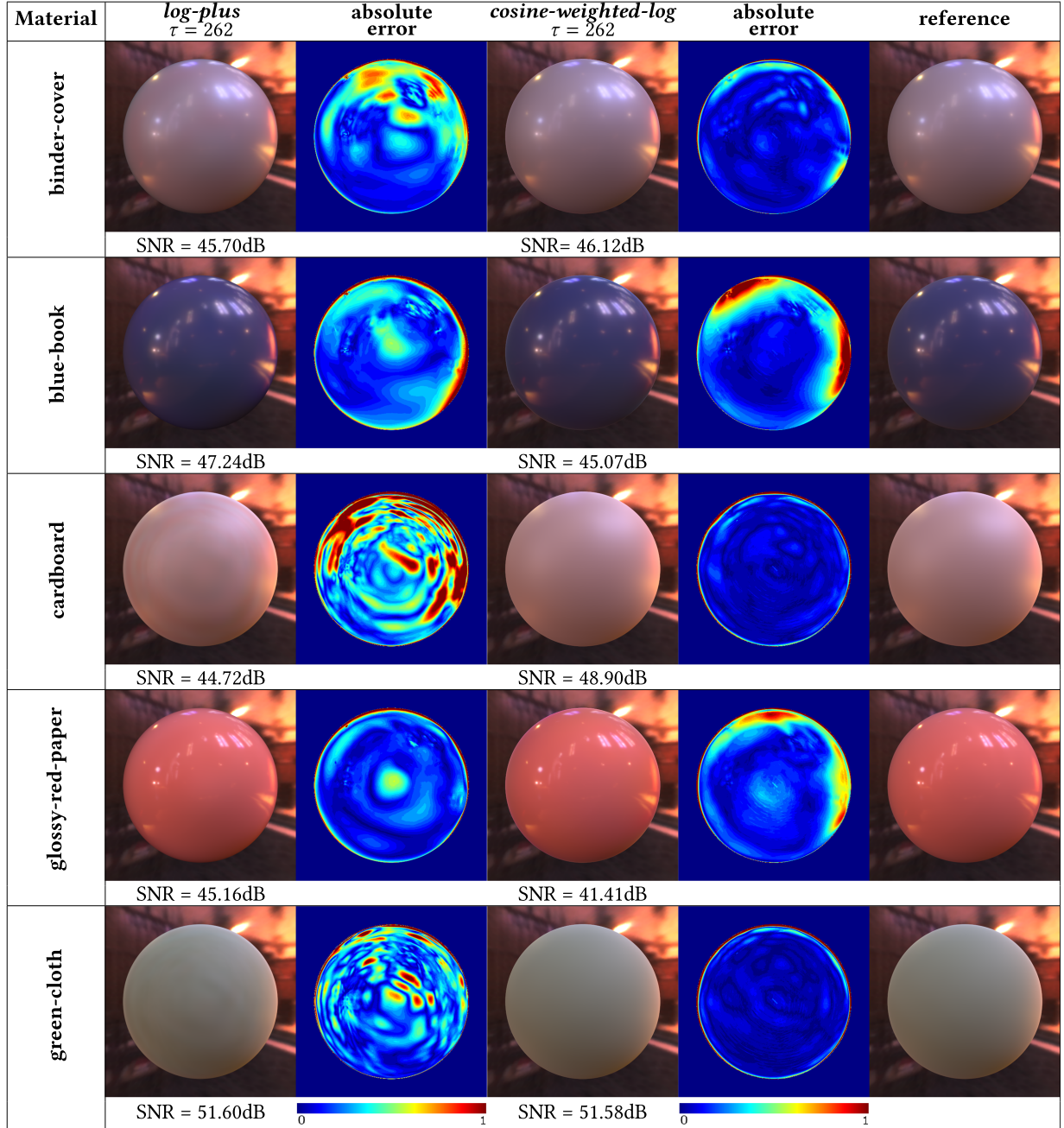


Fig. 3. Visual quality of selected materials from the DTU dataset [Nielsen et al. 2015]. For our model, we used $\tau_t = 262$ coefficients with *log-plus* and *cosine-weighted-log* transformations. Renderings are gamma corrected for display. The error images have been multiplied by 10.0 for visual comparisons. All images have been rendered using PBRT [Pharr and Humphreys 2010] with the *Grace Cathedral* environment map.

rendering, we invert Equations (1) and (2) to transform the reconstructed BRDFs back to the original linear domain, which is required by the renderer.

We compare our results to Bagher et al. [2016] (naive model), Bilgili et al. [2011] (Tucker decomposition) and Tongbuasirilai et al. [2019] (rank-1 CPD decomposition with $L = 1$) on 15 MERL test materials. The method of Bagher et al. [2016] stores

$(90+90+180+2) = 362$ coefficients per channel, Bligili et al. [2011] uses $(128+16+16+64+2) = 226$ coefficients, and the CPD decompositions from Tongbuasirilai et al. [2019] uses $(90+90+180) = 360$ coefficients per channel. Since the Tucker and CPD methods use an iterative approach, we limit our comparisons to $L = 1$, i.e., a single factorization was performed so that the number of coefficients used for all models were roughly the same. The CPD method was

Table 4. Rendering SNR and BRDF-space RAE Values Obtained with Our BRDF Model and That of Bagher et al. [2016], on 15 Test Materials of the MERL Dataset Using $\tau_t = 262$

Material	SNR (dB)		RAE	
	Our	Bagher	Our	Bagher
blue-fabric	62.16	63.88	0.8695	0.3596
blue-metallic-paint	60.53	44.86	0.4287	0.2736
dark-red-paint	54.80	57.89	0.2717	0.4955
gold-metallic-paint2	48.29	29.77	0.0811	0.7007
green-metallic-paint2	57.48	51.04	0.0726	0.4209
light-red-paint	51.68	51.36	0.2583	0.5639
pink-fabric2	52.71	52.44	0.8581	0.3388
purple-paint	47.22	44.29	0.1144	0.4428
red-fabric	55.05	51.44	0.7160	0.4163
red-metallic-paint	52.70	34.06	0.0752	0.7181
silver-metallic-paint2	44.55	27.11	0.6256	0.3626
specular-green-phenolic	53.09	35.69	0.0673	0.6414
specular-violet-phenolic	50.51	37.76	0.0592	0.6592
specular-yellow-phenolic	46.81	28.14	0.0683	0.7370
violet-acrylic	50.07	31.68	0.0653	0.5156

These materials were not used in our training set. Higher rendering SNR is highlighted in bold.

Table 5. Rendering SNR and BRDF-space RAE Values Obtained with Our BRDF Model on 5 Test Materials of the RGL-EPFL Dataset Using $\tau_t = 262$

Material	Our ρ_{t1}		Our ρ_{t2}		Sel.
	SNR (dB)	RAE	SNR (dB)	RAE	
acrylic-felt-green-rgb	43.23	0.9922	45.91	0.3104	45.91
cc-amber-citrine-rgb	26.33	0.5394	26.24	0.8420	26.24
ilm-l3-37-dark-green-rgb	38.92	0.9558	43.45	0.6060	43.45
paper-blue-rgb	38.92	0.9871	40.23	0.4416	40.23
vch-dragon-eye-red-rgb	40.48	0.8973	38.16	0.7989	38.16
Average	37.57	0.8744	38.80	0.5998	38.80

The bottom row shows mean of each column. The last column presents SNR results of our model selection method based on Gamma-mapped-MSE described in Section 3.4.

tested using two different parameterizations: the PDV [Löw et al. 2012; Tongbuasirilai et al. 2019] and HD [Rusinkiewicz 1998] parameterizations.

To the best of our knowledge, the model of Bagher et al. [2016] is the current state-of-the-art for non-parametric BRDF models. Therefore, we set the number of coefficients τ_t for our model such that we match the storage complexity of Bagher et al. [2016]. Since our representation is sparse, we require $1 + 1 + 1$ bytes for storing the location of each nonzero element in a sparse coefficient tensor with size $90 \times 90 \times 180$, as well as 8 bytes for each nonzero value. Simple calculations show that by using $\tau_t = 262$ coefficients for our model for each color channel, we can match the storage complexity of Bagher et al. [2016], which uses 362 coefficients to model each color channel of a BRDF. Note that the discrepancy in the number of coefficients is due to the fact that we need to store the location of nonzero values, which penalizes our method to use less coefficients.

For the rendered images, shown in Figures 2, 3, and 9, we apply gamma-corrected tone-mapping. The error images, also known as false-color, produced by normalizing the absolute image-space error in the range $[0, 1]$, followed by applying a *jet* color map using MATLAB. Normalization is done individually for each BRDF

Table 6. Rendering SNR and BRDF-space RAE Values Obtained with Our BRDF Model, on 8 Materials from the DTU Data Set [Nielsen et al. 2015] Using $\tau_t = 262$

Material	Our ρ_{t1}		Our ρ_{t2}		Sel.
	SNR (dB)	RAE	SNR (dB)	RAE	
binder-cover	45.70	0.0611	46.12	0.0303	46.12
blue-book	47.24	0.0574	45.07	0.0258	45.07
cardboard	44.72	0.1468	48.90	0.3779	48.90
glossy-red-paper	45.16	0.0436	41.41	0.0288	45.16
green-cloth	51.60	0.1145	51.58	0.7713	51.58
notebook	43.39	0.1838	47.15	0.2805	47.15
painted-metal	46.95	0.0817	51.84	0.1240	51.84
yellow-paper	47.14	0.1289	49.04	0.4873	49.04
Average	46.49	0.1022	47.64	0.2658	48.11

The bottom row shows means of each column. The last column present SNR results of our model selection method based on Gamma-mapped-MSE described in Section 3.4.

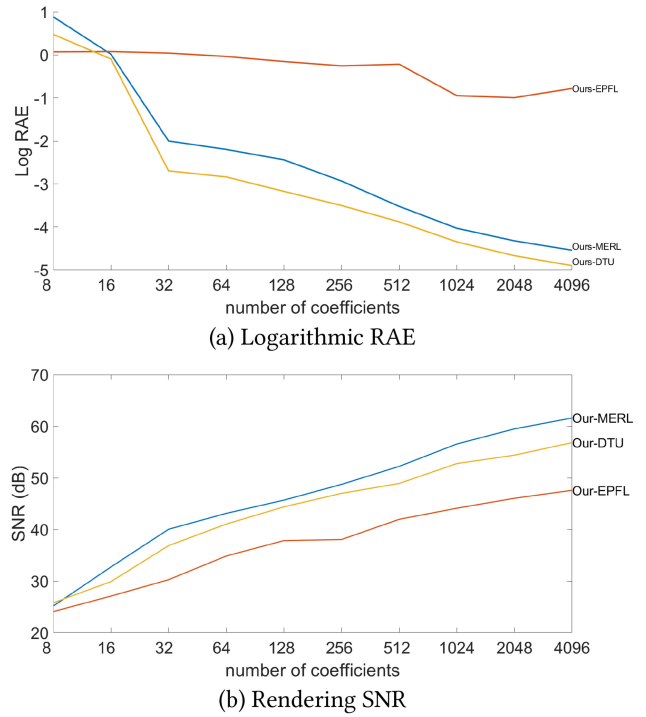


Fig. 4. (a) BRDF average error plots of all test materials from MERL, EPFL, and DTU data sets when reconstructed with increasing number of coefficients. The blue line represents *log-plus* transformation (ρ_{t1}) and the red line represents *cosine-weighted-log* transformation (ρ_{t2}). (b) Average rendering SNR plots of all test materials from MERL, EPFL, and DTU data sets when reconstructed with increasing number of coefficients. The plots show increasing trend when the number of coefficients increased.

represented using different models. All the error images are multiplied by 10 to facilitate visual comparisons. Apart from what follows, additional quantitative and qualitative results are included in the supplementary materials.

Table 3 reports SNR statistics for 15 test materials in the MERL database. The average SNR of our model is about 8dB, 5dB, and 10dB higher for *log-plus*, *cosine-weighted-log* and the proposed

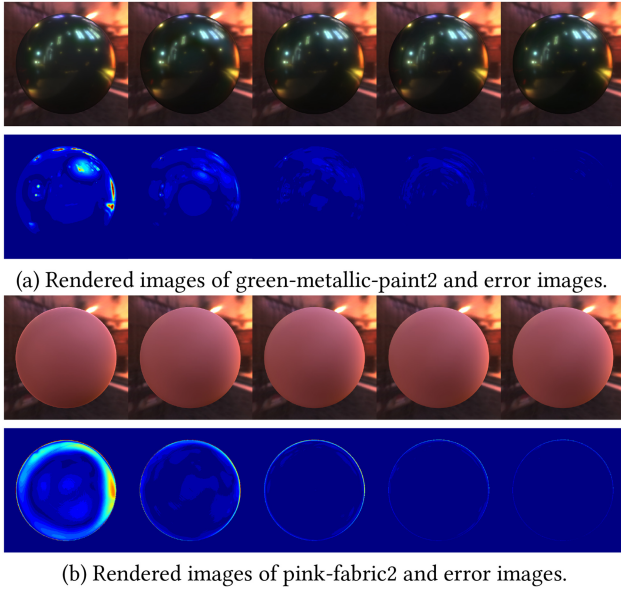


Fig. 5. The renderings, from left to right, were produced with τ_t equal to 16, 64, 256, 1,024, and 4,096, respectively. The second rows of panels (a) and (b) show error images using the Jet color map. All images have been rendered using PBRT [Pharr and Humphreys 2010] with the *Grace Cathedral* environment map.

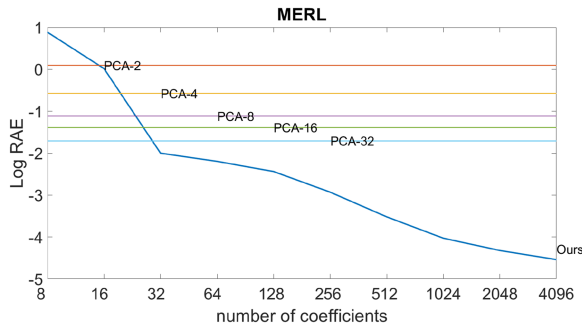


Fig. 6. RAE plots of MERL test materials when reconstructed with increasing number of coefficients. Each horizontal line, with PCA-x, indicates RAE from PCA-based technique [Nielsen et al. 2015], where x is the number of principal components.

model selection based on Gamma-mapped-MSE, respectively, when compared to Bagher et al.; moreover, our results show a smaller standard deviation on SNR. Additionally, our proposed model selection method can achieve higher SNR on average compared to the cases where we only use *log-plus* or *cosine-weighted-log*. The lower standard deviation indicates that the proposed model can represent the MERL materials more faithfully. Table 4 shows a direct comparison of our model to that of Bagher et al. [2016] for each BRDF in the MERL test set using rendering SNR and BRDF-space RAE. Here, we use our Gamma-mapped-MSE metric to choose between the transformations. Compared to the model of Bagher et al., our approach achieves significantly higher visual quality on 13 of 15 materials. In Figure 2, we present example

renderings of four BRDFs in the MERL test set modeled using our method and Bagher et al. [2016]. Our results are obtained using the proposed Gamma-mapped-MSE for model selection. The HDR environment map used here is *Grace Cathedral*.

The *cosine-weighted-log* transformation suppresses the grazing angle BRDF values. For diffuse materials, this leads to better visual results and significantly higher rendering SNR. It is evident from Figure 2 that the *log-plus* transformation is better for glossy materials as the *cosine-weighted-log* transformation leads to color artifacts for some materials, e.g., *gold-metallic-paint2*, *red-metallic-paint*, and *violet-acrylic*. More results for further analysis is available in the supplementary material.

Table 5 below shows rendering SNR and BRDF-space RAE values for the RGL-EPFL test set using both ρ_{t1} and ρ_{t2} . Our BRDF model and selection method can efficiently represent the RGL-EPFL data set with an average SNR of more than 38dB. Our model selection method on the RGL-EPFL test set missed on 2 of 5 materials, which are *cc-amber-citrine-rgb* and *vch-dragon-eye-red-rgb*. The SNR values demonstrate that our data-driven model can accurately represent and faithfully reconstruct the BRDFs in the test set. See the supplementary materials for rendered images obtained using our model applied on the RGL-EPFL data set. Although we use a fixed number of coefficients for all materials here for the sake of comparisons, i.e., $\tau_t = 262$, ideally we would like to tune this parameter for each BRDF individually. For instance, for challenging materials such as *cc-amber-citrine-rgb*, the model accuracy significantly increases with higher number of coefficients. We analyze the effect of τ_t on rendering SNR and BRDF RAE in Figure 4. Rendering results for two example BRDFs when we set τ_t to 16, 64, 256, 1,024, and 4,096 are included in Figure 5.

To demonstrate the robustness of our sparse non-parametric model for representing BRDFs in a test set, we also evaluate it using eight test samples in the DTU data set [Nielsen et al. 2015]. Note that we use the same dictionary described above and that none of the materials from the DTU data set were used in the training set. The results are summarized in Table 6, where we report rendering SNR and BRDF-space RAE for ρ_{t1} , ρ_{t2} , and our model selection based on *Gamma-mapped-MSE*. Our BRDF model and selection method can reproduce the DTU data set with average SNR of more than 48dB. Our model selection algorithm on the DTU test set missed on 2 of 8 materials, which are *blue-book* and *green-cloth*. Visual quality examples of the rendered images are presented in Figure 3. The difference between ρ_{t1} and ρ_{t2} is evident in this figure. We can see that ρ_{t1} is favored by glossy materials, while ρ_{t2} is more effective in modeling low-frequency or diffuse-like materials. For instance, we observe that the artifacts seen on the *card-board* and *green-cloth* renderings for *log-plus* (ρ_{t1}) do not appear in *cosine-weighted-log* (ρ_{t2}) renderings. For this figure, we used the *Grace Cathedral* HDR environment map.

Our results confirm the discrepancy between BRDF-space error metrics (such as RAE) and rendering quality using SNR. For example, *blue-metallic-paint* in Table 4, *cardboard* in Table 6, and *vch-dragon-eye-red-rgb* in Table 5 demonstrate how RAE contradicts the rendering SNR. The lower the BRDF-space RAE is, the more accurate the model represents a BRDF. However, a rendered image is dependent on a variety of additional factors such as geometry of objects, lighting environment, and viewing position. As a result,

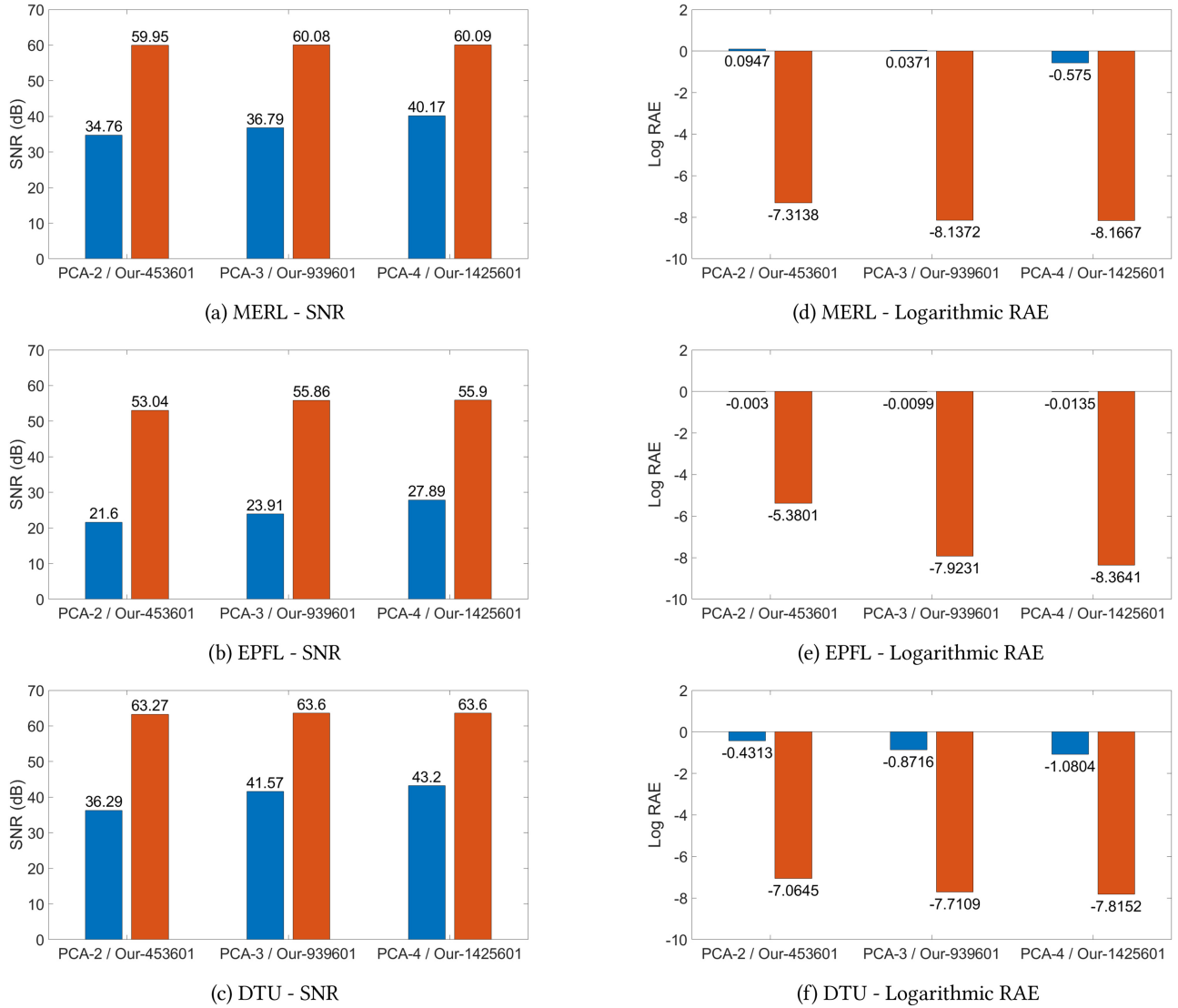


Fig. 7. Bar plots of average rendering SNR results from (a) MERL, (b) EPFL, and (c) DTU data sets. The blue bars are rendering SNR from PCA-based reconstructions while the orange bars are rendering SNR from our reconstructions. PCA-2, PCA-3, and PCA-4 are reconstructed results from PCA using 2, 3, and 4 principal components, respectively. Our-453601, Our-939601, and Our-1425601 are reconstructed results from our model using $\tau_t = 453,601$, $939,601$, and $1,425,601$, respectively.

the BRDF-space RAE and rendering SNR have to be considered together for the evaluation of a BRDF model.

Figure 4(a) demonstrates the effectiveness of our BRDF model on all data sets with respect to the number of coefficients, τ_t . In terms of the decline of error with respect to the number of coefficients, all data sets show a similar behavior. Moreover, all the data sets have poor RAE when $\tau_t < 32$. This behavior is expected, since the training sparsity is set to $\tau_l = 32$, $\tau_l = 64$, $\tau_l = 128$, and $\tau_l = 256$. And when we have $\tau_t < \tau_l$, we expect a decline in performance. Figure 4(b) indicates that our model generally produces incrementally better rendering results when the number of coefficients is increased for all the data sets. Our model performs best for the MERL data set, because most of the training data are from

MERL, while EPFL has fewer number of materials in the training. Although we do not include any materials from the DTU data set in the training, the rendering SNR and BRDF RAE are competitive with MERL reconstructions.

We also evaluated our BRDF model with the PCA-based method presented in Nielsen et al. [2015]; see Figure 6. Our RAE performs similar to PCA-based method with 32 principal components and when we have $\tau_t = 32$. We see that for a lower number of coefficients the PCA-based method outperforms our method, since our minimum training sparsity is 32. It should be noted that the PCA dictionary exhibits a very high storage cost compared to our BRDF dictionary ensemble. The PCA dictionary contains $1,458,000 \times p$ elements, where p is the number of principal

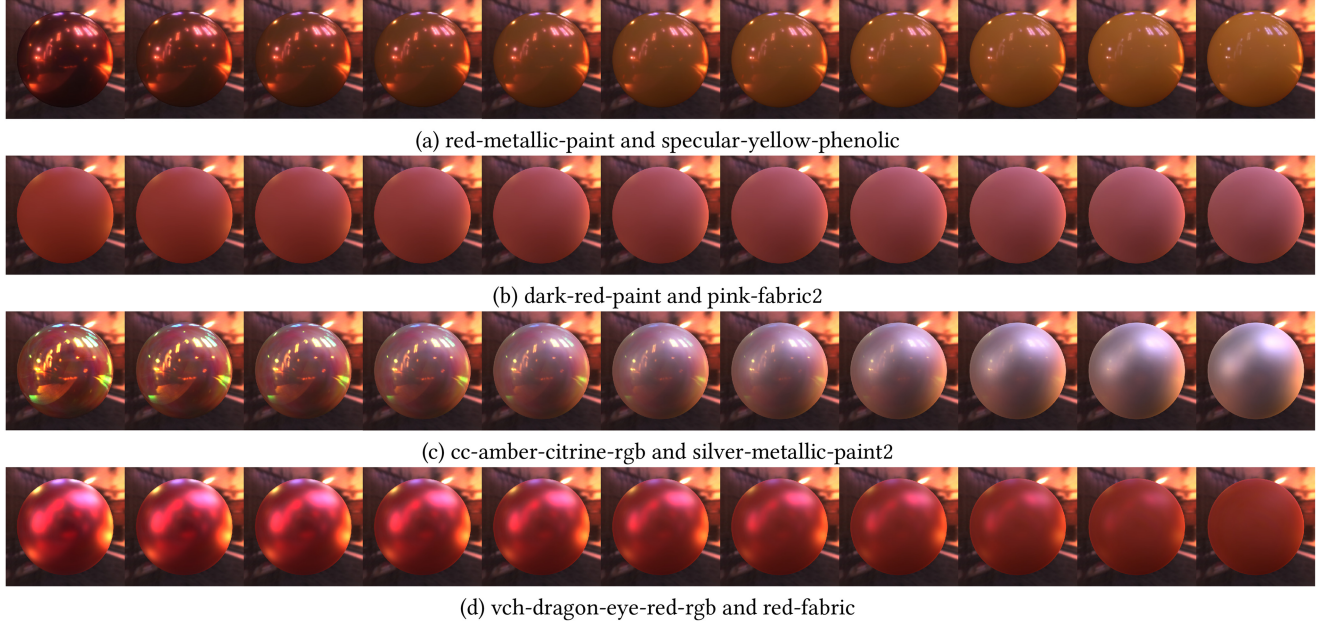


Fig. 8. Four examples of BRDF interpolations. We provide interpolation examples from different material classes. These examples are (a) glossy-glossy materials, (b) glossy-diffuse materials, (c) layered-glossy materials, and (d) layered-diffuse materials. From left to right, the interpolation coefficient α has values 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. The renderings were produced with $\tau_t = 512$.

components, while our combined dictionary ensemble consists of $(90 * 90 + 90 * 90 + 180 * 180) * 32 = 1,555,200$ elements. This shows that the storage complexity of the PCA dictionary is proportional to the number of principal components, whereas for our dictionary ensemble the storage complexity is constant. In Figures 7(a), 7(b), and 7(c), we present the rendering SNR and in Figures 7(d), 7(e), and 7(f), we present the logarithmic BRDF-space RAE for all three data sets in comparison to Nielsen et al. [2015], where we compensate for the size of the PCA dictionary by increasing the number of coefficients of our method, τ_t . When considering the size of the dictionaries, the PCA method with 2,3 and 4 principal components corresponds to our model with $\tau_t = 453,601$, $\tau_t = 939,601$, and $\tau_t = 1,425,601$, respectively. The bar plots indicate that, for all data sets, our model performs better than the PCA technique by a large margin on both quality metrics.

To demonstrate our BRDF interpolation method on the sparse coefficient space (the *sparseland*), we provide a number of challenging examples in Figure 8 for different material classes using Equation (19) introduced in Section 3.5. The interpolation coefficient, α , was set between 0.0 and 1.0 with increments of 0.1. Figure 8(a) illustrates the interpolation between two materials, i.e., *red-metallic-paint* and *specular-yellow-phenolic*, and Figure 8(b) shows the interpolation of two diffuse materials, namely, *dark-red-paint* and *pink-fabric2*. Figure 8(c) gives an example of a layered material, *cc-amber-citrine-rgb*, and a glossy material, *silver-metallic-paint2*. Finally, Figure 8(d) is an example of vinyl film, *vch-dragon-eye-red-rgb*, and a diffuse material, *red-fabric*. Note that each BRDF pair we use here share the same transformation (i.e., ρ_{t1} or ρ_{t2} as discussed in Section 3.1). As shown in the rendered images, our BRDF interpolation on the sparse coefficient space provides a smooth

transition between the material pairs. For these experiments, we used $\tau_t = 512$ for the model selection algorithm. Moreover, as mentioned in Section 3.5, we run the projection step of the model selection algorithm on the interpolated BRDF to obtain a desired sparsity, in this case $\tau_t = 512$. Note that this parameter is user-defined and can be changed to obtain a trade-off between quality and storage complexity for the interpolated BRDF.

In Figure 9, we evaluate our BRDF model using the *Princeton* scene with the following materials: *blue-metallic-paint*, *gold-metallic-paint2*, *pink-fabric2*, *silver-metallic-paint2*, and *specular-yellow-phenolic*. We rendered the scene with path tracing in PBRT [Pharr and Humphreys 2010] using the *uffizi* environment map and with 2^{17} samples-per-pixel. Figures 9(a) and 9(c) present rendered images from our model and Bagher et al., respectively. Our model achieves an 8.03dB advantage in SNR over the model of Bagher et al.

5 LIMITATIONS AND FUTURE WORK

The experiments conducted in this article are limited to isotropic BRDFs. This is partially due to the lack in availability of anisotropic BRDF data. As part of the future work, we aim to extend our sparse models to efficiently represent also anisotropic materials. Moreover, we acknowledge the fact that the discrepancy between BRDF-space errors and the rendering quality is still an open problem. Although we showed significant improvements using our Gamma-mapped-MSE, we believe that a more sophisticated metric that takes into account the support of the BRDF function can improve our results. Our model is relatively robust to noise. However, we believe that an application of a denoising pass that is

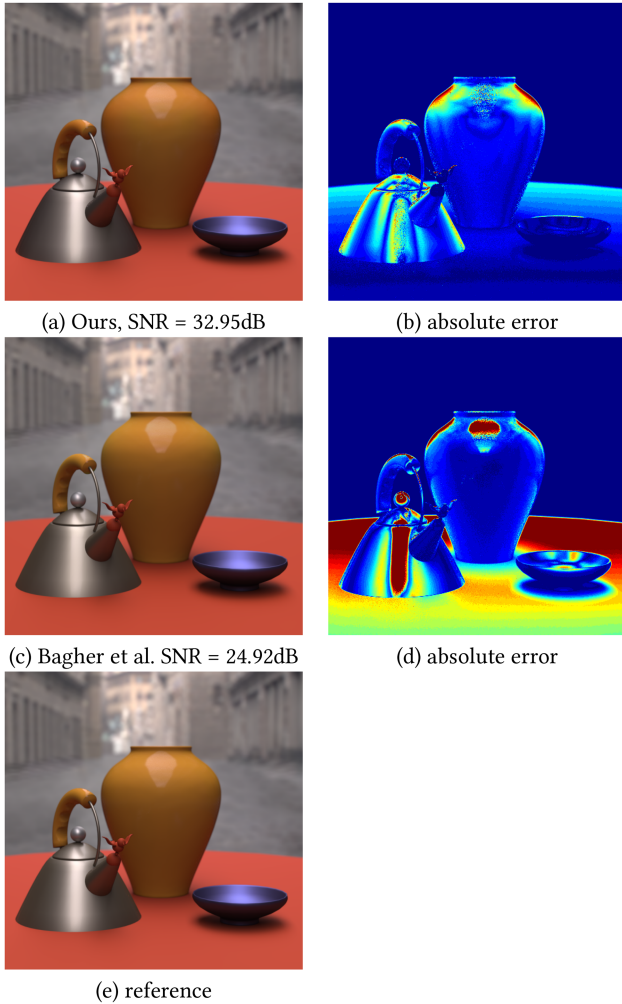


Fig. 9. Renderings of the *Princeton* scene using (a) our BRDF model and (c) the model of Bagher et al. All images were rendered at 131,072 samples/pixel using the path tracing algorithm of PBRT.

tailored to measured BRDFs, prior to training and model selection, can greatly improve our results. This is expected, since it is well-known that even a moderate amount of noise in measured BRDFs translates to lower rendering quality; and that noise reduces the sparsity of the representation, hence increasing the model complexity. An alternative to applying a denoiser is to modify the training and model selection methods to be noise-aware.

The proposed interpolation algorithm enables a variety of applications where the BRDFs processing can be performed directly in the coefficient space rather than the substantially higher-dimensional BRDF space. For instance, rendering of smooth surfaces where each point on the surface has a distinct measured BRDF (i.e., spatially varying BRDFs) can be efficiently performed using our model. Moreover, editing of measured BRDFs can be performed directly in the coefficient space by utilizing the shallow network proposed in Hu et al. [2020] that maps our sparse representation of materials to known BRDF properties. We have left these applications of our model for future work. Another

interesting venue for future work is the utilization of our model for sampling BRDFs, using, e.g., a gonioreflectometer, where only a few samples are taken to reconstruct a full BRDF, similar to Nielsen et al. [2015].

Our model is GPU-friendly, because it consists of parallel operations where atoms and coefficients can be efficiently distributed and operated among GPU cores. There exists a variety of GPGPU algorithms for performing key tasks in our method; for instance, a GPU-based implementation of the n -mode product is described in Baravdish et al. [2019]. As the main focus in this article is to derive and evaluate a novel non-parametric BRDF model, we have left the GPU implementation of our method to future work. However, given the small memory footprint of our model, and fast BRDF reconstruction using Equation (7), existing real-time ray tracing systems can be modified to accommodate our model for measured materials. Moreover, since the quality of our model is solely controlled using the τ_t parameter, one can linearly control the rendering performance using this parameter.

Quality metrics for effective BRDF fitting is an open problem [Lavoué et al. 2021]. In this article, we showed the discrepancy between the results of RAE and rendering SNR. We then propose a model selection technique to reduce the gap between the visual quality and the numerical quality metrics. We believe that our model selection algorithm can benefit from more accurate error metrics to further increase the visual quality. Due to the robustness of our model selection algorithm, any new mathematical error metric that is suitable for measured BRDFs can be used in Equation (6).

One of the limitations of the proposed model is the assumption of same resolution for the BRDFs in the training and testing sets. This is a limiting factor for BRDF data sets that are not densely sampled. One possible solution for these scenarios is that we train the model based on densely sampled BRDFs. For a given sparsely sampled BRDF to be modeled, we first create a dense BRDF by copying known values and setting unknown values to zero. We then iteratively perform the model selection algorithm on the obtained BRDF. At the first iteration, we obtain a crude approximation for the unknown values. This crude approximation is then improved iteratively by the consecutive runs of the model selection algorithm. We have left a full exploration of such method for future work.

6 CONCLUSIONS

This paper presented a novel non-parametric sparse BRDF model in which a measured BRDF is represented using a trained multi-dimensional dictionary ensemble and a set of sparse coefficients. We showed that with careful model selection over the space of multidimensional dictionaries and various BRDF transformations, we achieve significantly higher rendering quality and model accuracy compared to current state-of-the-art. Finally, we introduced a novel algorithm for interpolation of two or more BRDFs, modeled using our approach, directly in the coefficient space, even when the given BRDF data points used for the interpolation utilize distinct dictionaries. We evaluated the performance of our model and algorithm using three different data sets, MERL, RGL-EPFL, and one provided by Nielsen et al. [2015]. For the vast majority of the BRDFs used in the test set, we achieve a significant advantage over previous models.

REFERENCES

- Michal Aharon, Michael Elad, and Alfred Bruckstein. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* 54 (12 2006), 4311–4322. <https://doi.org/10.1109/TSP.2006.881199>
- Michael Ashikhmin and Peter Shirley. 2000. An anisotropic phong BRDF model. *J. Graph. Tools* 5, 2 (Feb. 2000), 25–32. <https://doi.org/10.1080/10867651.2000.10487522>
- Mahdi M. Bagher, John Snyder, and Derek Nowrouzezahrai. 2016. A non-parametric factor microfacet model for isotropic BRDFs. *ACM Trans. Graph.* 35, 5, Article 159 (July 2016), 16 pages. <https://doi.org/10.1145/2907941>
- Gabriel Baravdish, Ehsan Mianjidi, and Jonas Unger. 2019. GPU accelerated sparse representation of light fields. In *Proceedings of the 14th International Joint Conference on Computer Vision (VISAPP'19)*. INSTICC, SciTePress, 177–182. <https://doi.org/10.5220/0007393101770182>
- Pascal Barla, Laurent Belcour, and Romain Pacanowski. 2015. In praise of an alternative BRDF parametrization. In *Proceedings of the Workshop on Material Appearance Modeling*. <https://hal.inria.fr/hal-01172118>
- Aner Ben-Artzi, Ryan Overbeck, and Ravi Ramamoorthi. 2006. Real-time BRDF editing in complex lighting. *ACM Trans. Graph.* 25, 3 (July 2006), 945–954. <https://doi.org/10.1145/1141911.1141979>
- James C. Bieron and Pieter Peers. 2020. An adaptive brdf fitting metric. *Comput. Graph. Forum* 39, 4 (July 2020). <https://doi.org/10.1111/cgf.14054>
- Ahmet Bilgili, Aydn Öztürk, and Murat Kurt. 2011. A general BRDF representation based on tensor decomposition. *Comput. Graph. Forum* 30, 8 (2011), 2427–2439.
- James F. Blinn. 1977. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.* 11, 2 (July 1977), 192–198. <https://doi.org/10.1145/965141.563893>
- Matthew Brand. 2002. Charting a manifold. In *Proceedings of the 15th International Conference on Neural Information Processing Systems (NIPS'02)*. MIT Press, Cambridge, MA, 985–992.
- Zhe Chen, Shohai Nobuhara, and Ko Nishino. 2021. Invertible neural BRDF for object inverse rendering. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021), 1–1. <https://doi.org/10.1109/TPAMI.2021.3129537>
- L. Claustres, M. Paulin, and Y. Boucher. 2003. BRDF measurement modelling using wavelets for efficient path tracing. *Comput. Graph. Forum* 22, 4 (2003). <https://doi.org/10.1111/j.1467-8659.00718.x>
- Robert Cook and Kenneth Torrance. 1982. A reflectance model for computer graphics. *ACM Trans. Graph.* 1 (01 1982), 7–24. <https://doi.org/10.1145/965161.806819>
- V. Cooper, J. Bieron, and P. Peers. 2021. Estimating homogeneous data-driven BRDF parameters from a reflectance map under known natural lighting. *IEEE Trans. Visual. Comput. Graph.* 01 (May 2021), 1–1. <https://doi.org/10.1109/TVCG.2021.3085560>
- Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. 2018. Single-image SVBRDF capture with a rendering-aware deep network. *ACM Trans. Graph.* 37, 4, Article 128 (July 2018), 15 pages. <https://doi.org/10.1145/3197517.3201378>
- Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. 2019. Flexible SVBRDF capture with a multi-image deep network. *Comput. Graph. Forum (Proceedings of the Eurographics Symposium on Rendering)* 38, 4 (July 2019). Retrieved from <http://www.sop.inria.fr/revs/Basilic/2019/DADB19>
- Xin Ding, Wei Chen, and Ian J. Wassell. 2017. Joint sensing matrix and sparsifying dictionary optimization for tensor compressive sensing. *IEEE Trans. Signal Process.* 65, 14 (2017), 3632–3646. <https://doi.org/10.1109/TSP.2017.2699639>
- Yue Dong. 2019. Deep appearance modeling: A survey. *Visual Inform.* 3, 2, Article 59 (2019), 9 pages. <https://doi.org/10.1016/j.visinf.2019.07.003>
- Zhao Dong, Bruce Walter, Steve Marschner, and Donald P. Greenberg. 2016. Predicting appearance from measured microgeometry of metal surfaces. *ACM Trans. Graph.* 35, 1, Article 9 (Dec. 2016), 13 pages. <https://doi.org/10.1145/2815618>
- Jonathan Dupuy and Wenzel Jakob. 2018. An adaptive parameterization for efficient material acquisition and rendering. *Trans. Graph. (Proceedings of SIGGRAPH Asia)* 37, 6 (Nov. 2018), 274:1–274:18. <https://doi.org/10.1145/3272127.3275059>
- Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A. Wilson, and Paul Debevec. 2009. Estimating specular roughness and anisotropy from second order spherical gradient illumination. *Comput. Graph. Forum* 28, 4 (2009), 1161–1170. Retrieved from <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01493.x>. <https://doi.org/10.1111/j.1467-8659.2009.01493.x>
- Abhijeet Ghosh, Wolfgang Heidrich, Shruthi Achutha, and Matthew O'Toole. 2010. A basis illumination approach to BRDF measurement. *Int. J. Comput. Vision* 90, 2 (2010), 183–197. <https://doi.org/10.1007/s11263-008-0151-7>
- John C. Gower and Garnt B. Dijkstra. 2004. *Procrustes Problems*. Oxford Statistical Science Series, Vol. 30. Oxford University Press, Oxford, UK.
- Simon Hawe, Matthias Seibert, and Martin Kleinstaubert. 2013. Separable dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), 438–445.
- Nicolas Holzschuch and Romain Pacanowski. 2017. A two-scale microfacet reflectance model combining reflection and diffraction. *ACM Trans. Graph.* 36, 4, Article 66 (July 2017), 12 pages. <https://doi.org/10.1145/3072959.3073621>
- Bingyang Hu, Jie Guo, Yanjun Chen, Mengtian Li, and Yanwen Guo. 2020. DeepBRDF: A deep representation for manipulating measured BRDF. *Comput. Graph. Forum* 39, 2 (2020). <https://doi.org/10.1111/cgf.13920>
- Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. 2014. A comprehensive framework for rendering layered materials. *ACM Trans. Graph.* 33, 4, Article 118 (July 2014), 14 pages. <https://doi.org/10.1145/2601097.2601139>
- Jan Kautz and Michael D. McCool. 1999. Interactive rendering with arbitrary BRDFs using separable approximations. In *Proceedings of the 10th Eurographics Conference on Rendering (EGWR'99)*. Eurographics Association, 247–260. <https://doi.org/10.2312/EGWR/EGWR99/247-260>
- Tamara G. Kolda and Brett W. Bader. 2009. Tensor decompositions and applications. *SIAM Rev.* 51, 3 (Aug. 2009), 455–500. <https://doi.org/10.1137/07070111X>
- Guillaume Lavoué, Nicolas Bonneel, Jean-Philippe Farrugia, and Cyril Soler. 2021. Perceptual quality of BRDF approximations: Dataset and metrics. *Comput. Graph. Forum* 40, 2 (May 2021). Retrieved from <https://hal.inria.fr/hal-03128383>
- Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. 2004. Efficient BRDF importance sampling using a factored representation. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 496–505. <https://doi.org/10.1145/1015706.1015751>
- Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*.
- Joakim Löw, Joel Kronander, Anders Ynnerman, and Jonas Unger. 2012. BRDF models for accurate and efficient rendering of glossy surfaces. *ACM Trans. Graph.* 31, 1, Article 9 (Feb. 2012), 14 pages. <https://doi.org/10.1145/2077341.2077350>
- Mahdi Mousavi, Kaveh Abhari, Paul Babyn, and Javad Alirezaie. 2014. An adaptive approach to learn overcomplete dictionaries with efficient numbers of elements. *IEEE Trans. Signal Process.* 62 (06 2014), 3272–3283. <https://doi.org/10.1109/TSP.2014.2324994>
- Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. 2003. A data-driven reflectance model. *ACM Trans. Graph.* 22, 3 (July 2003), 759–769. <https://doi.org/10.1145/882262.882343>
- Raazia Mazhar and Paul D. Gader. 2008. EK-SVD: Optimized dictionary design for sparse representations. In *Proceedings of the 19th International Conference on Pattern Recognition*. 1–4. <https://doi.org/10.1109/ICPR.2008.4761362>
- Ehsan Mianjidi, Saghi Hajisharif, and Jonas Unger. 2019. A unified framework for compression and compressed sensing of light fields and light field videos. *ACM Trans. Graph.* 38, 3, Article 23 (May 2019), 18 pages. <https://doi.org/10.1145/3269980>
- Subhadip Mukherjee, Rupam Basu, and Chandra Sekhar Seelamantula. 2015. ℓ_1 -K-SVD: A robust dictionary learning algorithm with simultaneous update. *Signal Process.* 123 (12 2015). <https://doi.org/10.1016/j.sigpro.2015.12.008>
- Fred E. Nicodemus, Joseph C. Richmond, Jack J. Hsia, Irving W. Ginsberg, and T. Limperis. 1977. Geometrical considerations and nomenclature for reflectance. In *Radiometry*. Jones and Bartlett Publishers, Inc., 94–145.
- Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. 2015. On optimal, minimal BRDF sampling for reflectance acquisition. *ACM Trans. Graph.* 34, 6 (Nov. 2015), 186:1–186:11. <https://doi.org/10.1145/2816795.2818085>
- Ko Nishino and Stephen Lombardi. 2011. Directional statistics-based reflectance model for isotropic bidirectional reflectance distribution functions. *J. Opt. Soc. Am. A* 28, 1 (Jan. 2011), 8–18. <https://doi.org/10.1364/JOSAA.28.000008>
- Romain Pacanowski, Oliver Salazar Celis, Christophe Schlick, Xavier Granier, Pierre Poulin, and Annie Cuyt. 2012. Rational BRDF. *IEEE Trans. Visual. Comput. Graph.* 18, 11 (2012), 1824–1835. <https://doi.org/10.1109/TVCG.2012.73>
- Matt Pharr and Greg Humphreys. 2010. *Physically Based Rendering: From Theory To Implementation* (2nd ed.). Morgan Kaufmann, San Francisco, CA.
- Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural BTF compression and interpolation. *Comput. Graph. Forum (Proceedings of Eurographics)* 38, 2 (Mar. 2019).
- Ravi Ramamoorthi and Pat Hanrahan. 2001. An efficient representation for irradiance environment maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01)*. Association for Computing Machinery, New York, NY, 497–500. <https://doi.org/10.1145/383259.383317>
- Florian Roemer, Giovanni Del Galdo, and Martin Haardt. 2014. Tensor-based algorithms for learning multidimensional separable dictionaries. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'14)*. 3963–3967. <https://doi.org/10.1109/ICASSP.2014.6854345>
- Fabiano Romeiro, Yuriy Vasilev, and Todd Zickler. 2008. Passive reflectometry. In *Proceedings of the 10th European Conference on Computer Vision (ECCV'08)*. Springer-Verlag, Berlin, 859–872.
- Szymon Rusinkiewicz. 1998. A new change of variables for efficient BRDF representation. In *Rendering Techniques (Eurographics)*, George Drettakis and Nelson L. Max (Eds.). Springer, 11–22.
- Cristian Rusu and Bogdan Dumitrescu. 2012. Stagewise K-SVD to design efficient dictionaries for sparse representations. *IEEE Signal Process. Lett.* 19, 10 (2012), 631–634. <https://doi.org/10.1109/LSP.2012.2209871>
- Cyril Soler, Mahdi M. Bagher, and Derek Nowrouzezahrai. 2015. Efficient and accurate spherical kernel integrals using isotropic decomposition. *ACM Trans. Graph.* 34, 5, Article 161 (Nov. 2015), 14 pages. <https://doi.org/10.1145/2797136>

- Cyril Soler, Kartic Subr, and Derek Nowrouzezahrai. 2018. A versatile parameterization for measured material manifolds. *Comput. Graph. Forum* 37, 2 (Apr. 2018), 135–144. <https://doi.org/10.1111/cgf.13348>
- Michael Stark, James Arvo, and Brian Smits. 2005. Barycentric parameterizations for isotropic BRDFs. *IEEE Trans. Visual. Comput. Graph.* 11 (Apr. 2005), 126–38. <https://doi.org/10.1109/TVCG.2005.26>
- Tiancheng Sun, Henrik Wann Jensen, and Ravi Ramamoorthi. 2018. Connecting measured BRDFs to analytic BRDFs by data-driven diffuse-specular separation. *ACM Trans. Graph.* 37, 6, Article 273 (Dec. 2018), 15 pages. <https://doi.org/10.1145/3272127.3275026>
- Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. 2021. Neural BRDF representation and importance sampling. *Comput. Graph. Forum* 40, 6 (2021), 332–346. <https://doi.org/10.1111/cgf.14335>
- Tanaboon Tongbuasirilai, Jonas Unger, Joel Kronander, and Murat Kurt. 2019. Compact and intuitive data-driven BRDF models. *Visual Comput.* 36 (May 2019), 855–872.
- Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. 2013. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Trans. Graph.* 32, 4, Article 109 (July 2013), 12 pages. <https://doi.org/10.1145/2461912.2461944>
- Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques (EGSR'07)*. Eurographics Association, 195–206. <https://doi.org/10.2312/EGWR/EGSR07/195-206>
- Gregory J. Ward. 1992. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.* 26, 2 (July 1992), 265–272.

Received 2 December 2021; revised 12 April 2022; accepted 24 April 2022