A Sparse Non-parametric BRDF Model
Tanaboon Tongbuasirilai, Jonas Unger, Christine Guillemot, Ehsan Miandji

To cite this version:
Tanaboon Tongbuasirilai, Jonas Unger, Christine Guillemot, Ehsan Miandji. A Sparse Non-parametric BRDF Model. ACM Transactions on Graphics, 2022, pp.1-18. 10.1145/3533427. hal-03654734

HAL Id: hal-03654734
https://hal.science/hal-03654734
Submitted on 28 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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

This paper presents a novel sparse non-parametric 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-EFPL BRDF databases. Experimental results show that the proposed approach results in about 9.75dB higher SNR on average for rendered images as compared to current state-of-the-art models.


Additional Key Words and Phrases: Rendering, Reflectance and shading models, Machine learning, Dictionary learning, Non-parametric BRDF model, BRDF interpolation

ACM Reference Format:

1 INTRODUCTION

The bidirectional reflectance distribution function [Nicolodems et al. 1977] describes how light scatters at the surfaces of a scene, depending on their material characteristics. The 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 on the other hand 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. PCA or 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 paper 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 in order 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 paper 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 paper 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 paper 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 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 paper, 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 $a_i$, $A_{i_1,i_2}$, $\mathcal{A}_{i_1,...,i_N}$, respectively. The determinant of A is denoted $\det(A)$. The $\ell_p$ norm of a vector $s$, for $1 \leq p \leq \infty$, is denoted by $\|s\|_p$. Frobenius norm is denoted $\|s\|_F$. The $\ell_0$ pseudo-norm of a vector, denoted $\|s\|_0$, defines the number of non-zero elements in the vector. The $n$-mode product of a tensor $\mathcal{S}$ and a matrix $U$ is denoted $\mathcal{S} \times_n U$.

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 Tongbuasiriilai 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 advancements 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 [Sztajman 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 [Marsoumi 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 paper 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 $(\theta_b, \theta_d, \phi_d)$, where $\theta_b, \theta_d$ are elevation angles of the half vector and the difference vector respectively, $\phi_d$ 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 $\theta_h$, 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; Tongbuasirilai et al. 2019]. In this paper, 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., [Löw et al. 2012]:

$$\rho_1(\omega_h, \omega_d) = \log(\rho(\omega_h, \omega_d) + 1),$$

where $\rho$ is the original BRDF value, $\rho_1$ is the transformed BRDF value, and $(\omega_h, \omega_d)$ 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. [Nielsen et al. 2015]; however, we exclude the denominator. We call this transformation cosine-weighted-log:

$$\rho_2(\omega_h, \omega_d) = \log(\rho(\omega_h, \omega_d) \times \text{cosMap}(\omega_h, \omega_d) + 1),$$

where $\text{cosMap}$ is a function that maps the inputs $(\omega_h, \omega_d)$ directions in the MERL encoding to $(\omega_h, \omega_d)$ in standard spherical coordinates, and calculates the weights as $\max(\cos(\theta_h) \times \cos(\theta_d), \epsilon)$, 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 (1) yields better results when compared to the cosine-weighted-log transformation in (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. Here we do not assume any specific parameterization and only require that all the BRDFs in $\{X^{(i)}\}_{i=1}^{N_i}$ have the same resolution. Moreover, as discussed in Section 3.1, we utilize two BRDF transformations, $\rho_1$ and $\rho_2$. As a result, the training set consists of two versions of each BRDF.

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

**Table 1. SNR of rendered images using the BRDF dictionaries trained with different dictionary sparsity levels: 32, 64, 128, and 256.**

<table>
<thead>
<tr>
<th>Material</th>
<th>Ensemble with $t_1 = 32$</th>
<th>Ensemble with $t_2 = 64$</th>
<th>Ensemble with $t_1 = 128$</th>
<th>Ensemble with $t_2 = 256$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\rho_1$ SNR(dB)</td>
<td>$\rho_2$ SNR(dB)</td>
<td>$\rho_1$ SNR(dB)</td>
<td>$\rho_2$ SNR(dB)</td>
</tr>
<tr>
<td>blue-fabric</td>
<td>53.9003</td>
<td>58.4393</td>
<td>57.0890</td>
<td>61.1925</td>
</tr>
<tr>
<td>blue-metallic-paint</td>
<td>54.8105</td>
<td>56.8017</td>
<td>52.4779</td>
<td>59.7930</td>
</tr>
<tr>
<td>dark-red-paint</td>
<td>44.1094</td>
<td>51.9677</td>
<td>45.8218</td>
<td>52.4098</td>
</tr>
<tr>
<td>gold-metallic-paint2</td>
<td>46.9514</td>
<td>38.6907</td>
<td>45.6783</td>
<td>36.0324</td>
</tr>
<tr>
<td>green-metallic-paint2</td>
<td>50.7108</td>
<td>41.8161</td>
<td>49.4635</td>
<td>39.3023</td>
</tr>
<tr>
<td>light-red-paint</td>
<td>41.4139</td>
<td>49.0505</td>
<td>43.7449</td>
<td>48.7451</td>
</tr>
<tr>
<td>pink-fabric2</td>
<td>44.8244</td>
<td>49.3862</td>
<td>48.5446</td>
<td>52.5484</td>
</tr>
<tr>
<td>purple-paint</td>
<td>43.8932</td>
<td>38.8859</td>
<td>42.2491</td>
<td>47.5648</td>
</tr>
<tr>
<td>red-fabric</td>
<td>47.5606</td>
<td>52.3038</td>
<td>50.9287</td>
<td>54.7831</td>
</tr>
<tr>
<td>red-metallic-paint</td>
<td>47.2351</td>
<td>40.3386</td>
<td>46.9943</td>
<td>38.4251</td>
</tr>
<tr>
<td>silver-metallic-paint2</td>
<td>40.3291</td>
<td>42.9256</td>
<td>44.0442</td>
<td>43.2292</td>
</tr>
<tr>
<td>specular-green-phenolic</td>
<td>48.4841</td>
<td>41.6432</td>
<td>47.3226</td>
<td>36.5157</td>
</tr>
<tr>
<td>specular-violet-phenolic</td>
<td>48.2384</td>
<td>42.7994</td>
<td>47.4994</td>
<td>37.9801</td>
</tr>
<tr>
<td>specular-yellow-phenolic</td>
<td>46.4907</td>
<td>39.1758</td>
<td>44.5259</td>
<td>36.1666</td>
</tr>
<tr>
<td>violet-acrylic</td>
<td>48.7179</td>
<td>44.0610</td>
<td>48.7112</td>
<td>38.9536</td>
</tr>
<tr>
<td>Average</td>
<td>47.1779</td>
<td>45.8860</td>
<td>47.6730</td>
<td>45.5761</td>
</tr>
</tbody>
</table>

**Table 1. SNR of rendered images using the BRDF dictionaries trained with different dictionary sparsity levels: 32, 64, 128, and 256.**

<table>
<thead>
<tr>
<th>Material</th>
<th>Ensemble with $t_1 = 32$</th>
<th>Ensemble with $t_2 = 64$</th>
<th>Ensemble with $t_1 = 128$</th>
<th>Ensemble with $t_2 = 256$</th>
</tr>
</thead>
<tbody>
<tr>
<td>blue-fabric</td>
<td>53.9003</td>
<td>58.4393</td>
<td>57.0890</td>
<td>61.1925</td>
</tr>
<tr>
<td>blue-metallic-paint</td>
<td>54.8105</td>
<td>56.8017</td>
<td>52.4779</td>
<td>59.7930</td>
</tr>
<tr>
<td>dark-red-paint</td>
<td>44.1094</td>
<td>51.9677</td>
<td>45.8218</td>
<td>52.4098</td>
</tr>
<tr>
<td>gold-metallic-paint2</td>
<td>46.9514</td>
<td>38.6907</td>
<td>45.6783</td>
<td>36.0324</td>
</tr>
<tr>
<td>green-metallic-paint2</td>
<td>50.7108</td>
<td>41.8161</td>
<td>49.4635</td>
<td>39.3023</td>
</tr>
<tr>
<td>light-red-paint</td>
<td>41.4139</td>
<td>49.0505</td>
<td>43.7449</td>
<td>48.7451</td>
</tr>
<tr>
<td>pink-fabric2</td>
<td>44.8244</td>
<td>49.3862</td>
<td>48.5446</td>
<td>52.5484</td>
</tr>
<tr>
<td>purple-paint</td>
<td>43.8932</td>
<td>38.8859</td>
<td>42.2491</td>
<td>47.5648</td>
</tr>
<tr>
<td>red-fabric</td>
<td>47.5606</td>
<td>52.3038</td>
<td>50.9287</td>
<td>54.7831</td>
</tr>
<tr>
<td>red-metallic-paint</td>
<td>47.2351</td>
<td>40.3386</td>
<td>46.9943</td>
<td>38.4251</td>
</tr>
<tr>
<td>silver-metallic-paint2</td>
<td>40.3291</td>
<td>42.9256</td>
<td>44.0442</td>
<td>43.2292</td>
</tr>
<tr>
<td>specular-green-phenolic</td>
<td>48.4841</td>
<td>41.6432</td>
<td>47.3226</td>
<td>36.5157</td>
</tr>
<tr>
<td>specular-violet-phenolic</td>
<td>48.2384</td>
<td>42.7994</td>
<td>47.4994</td>
<td>37.9801</td>
</tr>
<tr>
<td>specular-yellow-phenolic</td>
<td>46.4907</td>
<td>39.1758</td>
<td>44.5259</td>
<td>36.1666</td>
</tr>
<tr>
<td>violet-acrylic</td>
<td>48.7179</td>
<td>44.0610</td>
<td>48.7112</td>
<td>38.9536</td>
</tr>
<tr>
<td>Average</td>
<td>47.1779</td>
<td>45.8860</td>
<td>47.6730</td>
<td>45.5761</td>
</tr>
</tbody>
</table>
\[ \{U^{(1,k)}, U^{(2,k)}, U^{(3,k)}\}_{k=1}^{K}, \text{ such that each BRDF, } X(i), \text{ can be decomposed as} \]
\[ X(i) = S^{(i)} X_1 U^{(1,k)} X_2 U^{(2,k)} X_3 U^{(3,k)}, \]
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 \( k \in \{1, \ldots, K\}. \) The notation \( X_n \) 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 \( \tau \) is a user-defined sparsity parameter to achieve a certain sparsity in the representation. Recall from Section 1 that the \( \ell_0 \) norm, denoted \( \|\cdot\|_0, \) counts the number of nonzero elements in a vector, a matrix, or a tensor. It is evident from (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 clusters, where each cluster is represented using one dictionary.

The ensemble training is performed by solving the following optimization problem
\[ \min_{U^{(j,k)}, S^{(j,k)}, M_{i,k}} \sum_{j=1}^{N_y} \sum_{k=1}^{K} M_{i,k} \|X(i) - \]
\[ S^{(j,k)} X_1 U^{(1,k)} X_2 U^{(2,k)} X_3 U^{(3,k)} \|_F^2 \] (4a)
subject to
\[ \left( U^{(j,k)} \right)^T U^{(j,k)} = I, \quad \forall k = 1, \ldots, K, \quad \forall j = 1, \ldots, 3, \] (4b)
\[ \|S^{(j,k)}\|_0 \leq \tau, \] (4c)
\[ \sum_{k=1}^{K} M_{i,k} = 1, \quad \forall i = 1, \ldots, N_l, \] (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 (4c), and the single dictionary representation is enforced by the constraint in (4d). The user-defined parameter \( \tau \) 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 representation. The supplementary document accompanying this manuscript contains the update rules for solving (4) iteratively with respect to \( U^{(j,k)}, S^{(j,k)}, \) and \( M_{i,k}. \)

Our sparse BRDF modeling is inspired by the Aggregate Multi-dimensional Dictionary Ensemble (AMDE) proposed by Miandji et al. [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, 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 \( 90 \times 90 \times 180 = 145800 \), 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. [Miandji et al. 2019], we perform multiple trainings of the same training set but with different values for the training sparsity parameter, \( \tau. \) 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; Turwattanapon 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 \( \tau \) is fixed for all the BRDFs in the training set, a small value for \( \tau \) will steer the optimization algorithm to more efficiently model low frequency (or diffuse-like) materials, while neglecting high frequency materials. Moreover, when \( \tau \) 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 \( \tau \) 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 \( \tau \) for a large and diverse set of BRDFs in the training set, we train multiple ensembles of dictionaries, each with a different value for \( \tau \), 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 paper, we train 4 ensembles, each with 8 dictionaries, which are trained with \( \tau \) = 32, \( \tau \) = 64, \( \tau \) = 128, and \( \tau \) = 256; hence, the final ensemble consists of 32 dictionaries. Note that the combined ensemble with 32 dictionaries only requires \( (32 \times (90 \times 90 + 90 \times 180 + 180 \times 180)) / 1024^2 = 5.93\text{MB} \) of storage, assuming 4 bytes for each element. Therefore, our ensemble can be used in applications where storage capabilities are very limited.
Table 2. Rendering SNR, Gamma-mapped-MSE, and MSE, obtained using our sparse BRDF model for \( \rho_1 \) and \( \rho_2 \). For each error metric, the best result between \( \rho_1 \) and \( \rho_2 \) 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 out of 15 materials. It can also be seen that MSE only selects the correct transformation for 3 out of 15 materials. For results generated using Gamma-mapped-MSE, we set \( \gamma = 2.0 \).

<table>
<thead>
<tr>
<th>Material</th>
<th>Rendering SNR (dB)</th>
<th>Gamma-mapped-MSE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \rho_1 )</td>
<td>( \rho_2 )</td>
<td>( \rho_1 )</td>
</tr>
<tr>
<td>blue-fabric</td>
<td>53.99</td>
<td>62.16</td>
<td>0.0233</td>
</tr>
<tr>
<td>blue-metallic-paint</td>
<td>51.65</td>
<td>60.53</td>
<td>0.0448</td>
</tr>
<tr>
<td>dark-red-paint</td>
<td>49.16</td>
<td>54.80</td>
<td>0.0616</td>
</tr>
<tr>
<td>gold-metallic-paint2</td>
<td>48.29</td>
<td>37.68</td>
<td>0.9248</td>
</tr>
<tr>
<td>green-metallic-paint2</td>
<td>57.48</td>
<td>43.36</td>
<td>0.8767</td>
</tr>
<tr>
<td>light-red-paint</td>
<td>46.51</td>
<td>51.68</td>
<td>0.0552</td>
</tr>
<tr>
<td>pink-fabric2</td>
<td>52.66</td>
<td>52.71</td>
<td>0.0230</td>
</tr>
<tr>
<td>purple-paint</td>
<td>45.81</td>
<td>47.24</td>
<td>0.1991</td>
</tr>
<tr>
<td>red-fabric</td>
<td>56.26</td>
<td>55.05</td>
<td>0.0177</td>
</tr>
<tr>
<td>red-metallic-paint</td>
<td>52.70</td>
<td>42.62</td>
<td>1.2910</td>
</tr>
<tr>
<td>silver-metallic-paint2</td>
<td>44.70</td>
<td>44.55</td>
<td>0.0988</td>
</tr>
<tr>
<td>specular-green-phenolic</td>
<td>53.09</td>
<td>36.67</td>
<td>0.9889</td>
</tr>
<tr>
<td>specular-violet-phenolic</td>
<td>50.51</td>
<td>38.21</td>
<td>0.9722</td>
</tr>
<tr>
<td>specular-yellow-phenolic</td>
<td>46.81</td>
<td>36.40</td>
<td>0.9454</td>
</tr>
<tr>
<td>violet-acrylic</td>
<td>50.07</td>
<td>42.61</td>
<td>0.7770</td>
</tr>
</tbody>
</table>

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.

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 \( \rho_1 \) and \( \rho_2 \).
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 \( N_t \gg K \).

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

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

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 (3) is defined by the storage complexity of the sparse coefficient tensor \( \mathbf{S}^{(i)} \) in (5). We store the nonzero elements in \( \mathbf{S}^{(i)} \) as tuples of nonzero element locations and values, denoted \( \{ l_1^{(i)}, l_2^{(i)}, l_3^{(i)} ; \hat{S}^{(i)} \}_{j=1}^{T_2} \), where the indices \( l_1^{(i)}, l_2^{(i)}, \) and \( l_3^{(i)} \) store the location of the \( t \)th nonzero element of \( \mathbf{S}^{(i)} \), while the corresponding value is \( \hat{S}^{(i)} \).

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

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

where \( k \) is the index of the dictionary chosen by the model selection method described above. Thanks to the fact that the coefficient tensor \( \mathbf{S}^{(i,k)} \) is sparse, Equation (7) is computationally tractable even for real-time applications. Indeed, we can evaluate (7) by only multiplying the nonzero elements of \( \mathbf{S}^{(i,k)} \) with the dictionary, as shown in [Miandji et al. 2019]. Reconstructing a single element in a BRDF only requires \( 3T \) operations of type multiply-add (MAD). Indeed, for ray tracing applications, we only need to reconstruct one or a few elements of the BRDF tensor at each ray intersection point. Moreover, since \( T_1 \) is user-defined, our method is computationally flexible and can provide an intuitive which makes it suitable for a variety of applications.

Since our dictionary is trained with two sets of transformed BRDFs, i.e. \( \rho_{t1} \) and \( \rho_{t2} \), 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 \( \rho_{t1} \) and \( \rho_{t2} \). 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. [Bieron and Peers 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 \( \rho_{t1} \) and \( \rho_{t2} \) variants of BRDFs. From our observations, we found that the MSE as an error metric in the model selection algorithm, see (6), does not match a selection method based on rendering quality to choose from \( \rho_{t1} \) and \( \rho_{t2} \), see Table 2. To address this problem, we use a Gamma mapping function, \( \Gamma(\rho, \gamma) = \rho^{1/\gamma} \), on the reference \( \rho_{t1} \) and \( \rho_{t2} \), prior to computing the MSE. We refer to this error metric as \textit{Gamma-mapped-MSE}. Note that since the reference BRDF is in linear BRDF domain, i.e. it is not transformed, we invert \( \rho_{t1} \) and \( \rho_{t2} \) according to (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 \( \rho_{t1} \) and \( \rho_{t2} \). 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 \( \rho_{t1} \) and \( \rho_{t2} \) with respect to rendering SNR for 13 out of 15 materials. The two exceptions are \textit{red-fabric} and \textit{silver-metallic-paint2}. It can also be seen that MSE only selects the correct transformation for 3 out 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 \textit{Gamma-mapped-MSE}), BRDF transformations (using \( \rho_{t1} \) and \( \rho_{t2} \), as well as model complexity (via \( T_1 \)).

### 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; Matsukis 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; Matsukis et al. 2003], the local subspaces in our model can have different dimensionalties 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.

In order 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 Principal Component Analysis (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 \( B_1, B_2, B_3 \) be three BRDFs that respectively use the dictionaries \( \{U^{(1,1)}, U^{(2,1)}, U^{(3,1)}\}, \{U^{(1,1)}, U^{(2,1)}, U^{(3,1)}\}, \) and \( \{U^{(1,2)}, U^{(2,2)}, U^{(3,2)}\} \), respectively; i.e., we assume that \( B_1 \) and \( B_2 \) use the same dictionary from the ensemble. The linearly interpolated BRDF obtained from \( B_1 \) and \( B_2 \), which we denote \( I \), is then given by

\[
I = (1 - \alpha)B_1 + \alpha B_2
\]

\[
= (1 - \alpha)S_1 \times_1 U^{(1,1)} \times_2 U^{(2,1)} \times_3 U^{(3,1)} + \alpha S_2 \times_1 U^{(1,1)} \times_2 U^{(2,1)} \times_3 U^{(3,1)}
\]

\[
= ((1 - \alpha)S_1 + \alpha S_2) \times_1 U^{(1,1)} \times_2 U^{(2,1)} \times_3 U^{(3,1)},
\]

where \( S_1 \) and \( S_2 \) are the sparse coefficients of \( B_1 \) and \( B_2 \), respectively; the parameter \( \alpha \) is the interpolation coefficient. Note that, equation (10) follows from (9) only because the dictionary \( \{U^{(1,1)}, U^{(2,1)}, U^{(3,1)}\} \) is used by both \( B_1 \) and \( B_2 \). 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 (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 (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 \( B_1 \) and \( B_3 \), which use two distinct dictionaries \( \{U^{(1,1)}, U^{(2,1)}, U^{(3,1)}\} \) and \( \{U^{(1,3)}, U^{(2,3)}, U^{(3,3)}\} \), respectively. Our method is based on a simple observation: Given two orthonormal matrices, say \( D_1 = D_2R \) or \( D_2 = D_3R \), More concretely,

**Proposition 1.** Let \( A = \{A_1, A_2, A_3\} \) be two distinct orthonormal matrices. Then the solution of

\[
\min_R \|R - AR\|_F \quad \text{subject to} \quad R^T R = I,
\]

is \( R = A^T B_1 \).

The proof of Proposition 1 can be found in [Gower and Dijkstra 2004]. Note that Proposition 1 is a variant of the Orthogonal Procrustes problem with the difference that here we assume that the matrices \( A = \{A_1, A_2, A_3\} \) and \( B = \{B_1, B_2, B_3\} \) are orthonormal; as a result, the transformation matrix \( R \) is either a rotation or rotation-reflection matrix, depending on whether \( \det(R) = 1 \) or \( \det(R) = -1 \).

Let \( R^{(i,k)\rightarrow(i,j)} \) be the transformation matrix that transforms the orthonormal matrix \( U^{(i,k)} \) to \( U^{(i,j)} \); i.e., we have that \( U^{(i,j)} = U^{(i,k)} R^{(i,k)\rightarrow(i,j)} \). It should be noted that, according to Proposition 1, we can only compute \( R^{(i,k)\rightarrow(i,j)} \) for two orthonormal matrices of the same dimensionality. This is reflected in the fact that the index \( i \) is shared by \( U^{(i,k)} \) and \( U^{(i,j)} \), i.e., the matrices define an orthonormal basis for the \( i \)th mode of BRDFs. The transformation matrices, to be obtained using Proposition 1, between the dictionaries for \( B_1 \) and \( B_3 \) are

\[
U^{(1,2)} = U^{(1,1)} R^{(1,1)\rightarrow(1,2)}
\]

\[
U^{(2,2)} = U^{(2,1)} R^{(2,1)\rightarrow(2,2)}
\]

\[
U^{(3,2)} = U^{(3,1)} R^{(3,1)\rightarrow(3,2)}
\]

We can now state the formula for interpolating between two BRDFs that are not represented by a common dictionary, namely \( B_1 \) and \( B_3 \) as given by the example above.

\[
I = (1 - \alpha)B_1 + \alpha B_3
\]

\[
= (1 - \alpha)S_1 \times_1 U^{(1,1)} \times_2 U^{(2,1)} \times_3 U^{(3,1)} + \alpha S_3 \times_1 U^{(1,1)} \times_2 U^{(2,1)} \times_3 U^{(3,1)}
\]

\[
= ((1 - \alpha)S_1 + \alpha S_3) \times_1 U^{(1,1)} \times_2 U^{(2,1)} \times_3 U^{(3,1)}
\]

\[
= \left( (1 - \alpha)S_1 + \alpha S_3 \right) \times_1 U^{(1,1)} \times_2 U^{(2,1)} \times_3 U^{(3,1)},
\]

where (18) follows from (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 $B_2$ to transform them into the coordinate space of $B_1$ defined by $\{U^{(1,1)}, U^{(2,1)}, U^{(3,1)}\}$. 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 (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 (19).

It can be noted from (19) that $S_3$ is not necessarily sparse since it is obtained from multiplying the sparse tensor $S_3$ by the dense matrices $[R^{(1,1)} \rightarrow (1,2), R^{(2,1)} \rightarrow (2,2), R^{(3,1)} \rightarrow (3,2)]$. 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. $\{U^{(1,1)}, U^{(2,1)}, U^{(3,1)}\}$, 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 paper, we utilize the Rusinkiewicz’s parameterization defined by $\{\theta_b, \theta_d, \phi_d\}$. 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 $512 \times 512$ 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 $\tau_1 = 32$, $\tau_2 = 64$, $\tau_3 = 128$, and $\tau_4 = 256$. 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 ($\rho_{11}$) and cosine-weighted-log ($\rho_{22}$) functions before starting the training, hence resulting in 272 materials. The training took about 11 hours on a machine with 40 cores clocked at 2.4GHz. 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 3.8GHz.

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 [Miandji et al. 2019] to recover a single element. In addition, since the reconstruction in (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.

To evaluate our sparse BRDF model, we use two quality metrics: Signal-to-Noise Ratio (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 $\rho_{11}$ or $\rho_{22}$ depending on the model selection result) to transform it back to the linear space prior to computing the RAE. The RAE is defined as

\[
RAE = \sqrt{\frac{\sum(r_{ref} - r_{recon})^2}{\sum(r_{ref}^2)}}.
\]

where $r_{ref}$ is the reference BRDF, and $r_{recon}$ 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
Table 3. Average, standard deviation, minimum, and maximum rendering SNR values of each BRDF model obtained from 15 materials in the MERL dataset. 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).

<table>
<thead>
<tr>
<th>BRDF Model</th>
<th>Average SNR (dB)</th>
<th>Standard Deviation</th>
<th>Minimum SNR (dB)</th>
<th>Maximum SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours, $\tau = 262$, using only $\rho_1$</td>
<td>50.65</td>
<td>3.8106</td>
<td>44.70</td>
<td>57.49</td>
</tr>
<tr>
<td>Ours, $\tau = 262$, using only $\rho_2$</td>
<td>47.08</td>
<td>5.9319</td>
<td>36.40</td>
<td>62.16</td>
</tr>
<tr>
<td>Ours, $\tau = 262$, using Gamma-mapped-MSE</td>
<td>52.51</td>
<td>4.9752</td>
<td>44.55</td>
<td>62.16</td>
</tr>
<tr>
<td>Bagher et al. [2016]</td>
<td>42.76</td>
<td>11.6323</td>
<td>27.11</td>
<td>63.88</td>
</tr>
<tr>
<td>Bilgili et al. [2011]</td>
<td>32.63</td>
<td>5.8724</td>
<td>22.86</td>
<td>43.17</td>
</tr>
<tr>
<td>Tongbuasirilai et al. [2019] using CPD-PDV rank-1 (L=1)</td>
<td>33.83</td>
<td>5.5236</td>
<td>22.22</td>
<td>42.71</td>
</tr>
<tr>
<td>Tongbuasirilai et al. [2019] using CPD-HD rank-1 (L=1)</td>
<td>32.51</td>
<td>8.4995</td>
<td>22.97</td>
<td>52.27</td>
</tr>
</tbody>
</table>

Fig. 2. Visual quality of selected materials from the MERL test set. For our model we used $\tau = 262$ coefficients in order to compare with the method of Bagher et al. [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.

rendering SNR only takes into account a portion of the BRDF values, depending on the rendering setup. Note that for 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, Bilgili 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

<table>
<thead>
<tr>
<th>Material</th>
<th>log-plus (\tau = 262) absolute error</th>
<th>cosine-weighted-log (\tau = 262) absolute error</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>binder-cover</td>
<td><img src="image1.png" alt="Image" /> SNR = 45.70dB</td>
<td><img src="image2.png" alt="Image" /> SNR= 46.12dB</td>
<td></td>
</tr>
<tr>
<td>blue-book</td>
<td><img src="image3.png" alt="Image" /> SNR = 47.24dB</td>
<td><img src="image4.png" alt="Image" /> SNR = 45.07dB</td>
<td></td>
</tr>
<tr>
<td>cardboard</td>
<td><img src="image5.png" alt="Image" /> SNR = 44.72dB</td>
<td><img src="image6.png" alt="Image" /> SNR = 48.90dB</td>
<td></td>
</tr>
<tr>
<td>glossy-red-paper</td>
<td><img src="image7.png" alt="Image" /> SNR = 45.16dB</td>
<td><img src="image8.png" alt="Image" /> SNR = 41.41dB</td>
<td></td>
</tr>
<tr>
<td>green-cloth</td>
<td><img src="image9.png" alt="Image" /> SNR = 51.60dB</td>
<td><img src="image10.png" alt="Image" /> SNR = 51.58dB</td>
<td></td>
</tr>
</tbody>
</table>
Table 5. Rendering SNR and BRDF-space RAE values obtained with our BRDF model on 5 test materials of the RGL-EPFL dataset using $\tau_f = 262$. 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.

<table>
<thead>
<tr>
<th>Material</th>
<th>SNR (dB)</th>
<th>RAE</th>
<th>Our</th>
<th>Bagher</th>
<th>Our</th>
<th>Bagher</th>
</tr>
</thead>
<tbody>
<tr>
<td>acrylic-felt-green-rgb</td>
<td>43.23</td>
<td>0.9922</td>
<td>45.91</td>
<td>0.3104</td>
<td>45.91</td>
<td></td>
</tr>
<tr>
<td>cc-amber-citrine-rgb</td>
<td>26.33</td>
<td>0.5394</td>
<td>26.24</td>
<td>0.8420</td>
<td>26.24</td>
<td></td>
</tr>
<tr>
<td>ilm-l3-37-dark-green-rgb</td>
<td>38.92</td>
<td>0.9558</td>
<td>43.45</td>
<td>0.6060</td>
<td>43.45</td>
<td></td>
</tr>
<tr>
<td>paper-blue-rgb</td>
<td>38.92</td>
<td>0.9871</td>
<td>40.23</td>
<td>0.4416</td>
<td>40.23</td>
<td></td>
</tr>
<tr>
<td>vch-dragon-eye-red-rgb</td>
<td>40.48</td>
<td>0.8973</td>
<td>38.16</td>
<td>0.7989</td>
<td>38.16</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>37.37</td>
<td>0.8744</td>
<td>38.80</td>
<td>0.5998</td>
<td>38.80</td>
<td></td>
</tr>
</tbody>
</table>

were roughly the same. The CPD method was tested using two different parameterizations: the PDV [Löw et al. 2012; Tongbuasirilai et al. 2019] and HD [Ruskiewicz 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 $\tau_f$ for our model such that we match the storage complexity of [Bagher et al. 2016]. Since our representation is sparse, we require $1 + 1 + 2$ 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 $\tau_f = 262$ 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 represented using different models. All the error images are multiplied...
The posed model selection method can achieve higher SNR on average and show a smaller standard deviation on SNR. Additionally, our proposed model selection based on Gamma-mapped-MSE, respectively, when compared to Bagher et al.; moreover, our results about materials in the MERL database. The average SNR of our model is supplementary materials.

Additional quantitative and qualitative results are included in the supplementary materials.

Table 3 reports Signal-to-Noise Ratio (SNR) statistics for 15 test materials in the MERL database. The average SNR of our model is about 8dB, 5dB and 10dB higher for \textit{log-plus}, \textit{cosine-weighted-log} and the proposed 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 \textit{log-plus} or \textit{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 out 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 \textit{Grace Cathedral}.

The \textit{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 \textit{log-plus} transformation is better for glossy materials as the \textit{cosine-weighted-log} transformation leads to color artifacts for some materials, e.g. \textit{gold-metallic-paint2}, \textit{red-metallic-paint}, and \textit{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 $\rho_{11}$ and $\rho_{12}$. 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 out of 5 materials, which are \textit{cc-amber-citrine-rgb} and \textit{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. $\tau_1 = 262$, ideally we would like to tune this parameter for each BRDF individually. For instance, for challenging materials such as \textit{cc-amber-citrine-rgb}, the model accuracy significantly increases with higher number of coefficients. We analyze the effect of $\tau_1$ on rendering SNR and BRDF RAE in Figure 4. Rendering results for two example BRDFs when we set $\tau_1$ to 16, 64, 256, 1024, and 4096 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 8 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 $\rho_{11}$, $\rho_{12}$, 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 out of 8 materials, which are \textit{blue-book} and \textit{green-cloth}. Visual quality examples of the rendered images are presented in Figure 3. The difference between $\rho_{11}$ and $\rho_{12}$ is evident in this figure. We can see that $\rho_{11}$ is favored by glossy materials, while $\rho_{12}$ is more effective in modeling low-frequency or diffuse-like materials. For instance, we observe that the artifacts seen on the \textit{cardboard} and \textit{green-cloth} renderings for \textit{log-plus} ($\rho_{11}$) do not appear in \textit{cosine-weighted-log}.

![Rendered images of green-metallic-paint2 and error images.](image1)

![Rendered images of pink-fabric2 and error images.](image2)

Fig. 5. The renderings, from left to right, were produced with $\tau_1$ equal to 16, 64, 256, 1024, and 4096, respectively. The second rows of (a) and (b) show error images using the Jet color map. All images have been rendered using PBRT [Pharr and Humphreys 2010] with the \textit{Grace Cathedral} environment map.

![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.](image3)

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.

by 10 to facilitate visual comparisons. Apart from what follows, additional quantitative and qualitative results are included in the supplementary materials.

Table 3 reports Signal-to-Noise Ratio (SNR) statistics for 15 test materials in the MERL database. The average SNR of our model is about 8dB, 5dB and 10dB higher for \textit{log-plus}, \textit{cosine-weighted-log} and the proposed 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 \textit{log-plus} or \textit{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 out 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 \textit{Grace Cathedral}.

The \textit{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 \textit{log-plus} transformation is better for glossy materials as the \textit{cosine-weighted-log} transformation leads to color artifacts for some materials, e.g. \textit{gold-metallic-paint2}, \textit{red-metallic-paint}, and \textit{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 $\rho_{11}$ and $\rho_{12}$. 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 out of 5 materials, which are \textit{cc-amber-citrine-rgb} and \textit{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. $\tau_1 = 262$, ideally we would like to tune this parameter for each BRDF individually. For instance, for challenging materials such as \textit{cc-amber-citrine-rgb}, the model accuracy significantly increases with higher number of coefficients. We analyze the effect of $\tau_1$ on rendering SNR and BRDF RAE in Figure 4. Rendering results for two example BRDFs when we set $\tau_1$ to 16, 64, 256, 1024, and 4096 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 8 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 $\rho_{11}$, $\rho_{12}$, 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 out of 8 materials, which are \textit{blue-book} and \textit{green-cloth}. Visual quality examples of the rendered images are presented in Figure 3. The difference between $\rho_{11}$ and $\rho_{12}$ is evident in this figure. We can see that $\rho_{11}$ is favored by glossy materials, while $\rho_{12}$ is more effective in modeling low-frequency or diffuse-like materials. For instance, we observe that the artifacts seen on the \textit{cardboard} and \textit{green-cloth} renderings for \textit{log-plus} ($\rho_{11}$) do not appear in \textit{cosine-weighted-log}.
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, 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, $\tau_l$. 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 $\tau_l < 32$. 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 $\tau_l < \tau_l$, 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 $\rho_2$ renderings. For this figure we used the Grace Cathedral HDR environment map.

Figure 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-45601, Our-939601 and Our-1425601 are reconstructed results from our model using $\tau_l = 45601$, 939601 and 1425601 respectively.
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 $\tau = 32$. 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 $1458000 \times p$ elements, where $p$ is the number of principal components, while our combined dictionary ensemble consists of $(90 \times 90 + 90 \times 180 + 180) \times 32 = 1,555,200$ elements. This shows that the storage complexity of the PCA dictionary is proportional to the number of principal components, whereas our dictionary ensemble the storage complexity is constant. In figures 7(a), 7(b), and 7(c), we present the rendering SNR in and 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, $\tau$. When considering the size of the dictionaries, the PCA method with 2,3 and 4 principal components corresponds to our model with $\tau_2 = 453601$, $\tau_3 = 939601$, and $\tau_4 = 1425601$, 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, $\alpha$, 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. $p_1$ or $p_2$ 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 $\tau_p = 512$ 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 $\tau_p = 512$. 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.
Figure 9. Renderings of the Princeton scene using (a) our BRDF model and (c) the model of Bagher et al. All images were rendered at 131072 samples/pixel using the path tracing algorithm of PBRT.

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 paper 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 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, e.g., a gonio-reflectometer, 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 paper 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 Eq. (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 $\tau_0$ 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 paper, 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 (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 multidimensional 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.

7 ACKNOWLEDGEMENTS

This work was in part funded by EU H2020 Research, and Innovation Programme under Grant No.694122 (ERC advanced Grant CLIM).


