Data Driven BRDF Material Editing

Sachin Shah Sakshum Kulshrestha

Abstract

Designing 3D worlds is challenging for many and remains time consuming to perfect. One limiting factor is constructing realistic materials to give the world life. Existing material generation tools are limited in expressiveness. Measured bidirectional reflectance distribution functions (BRDFs) offer a unique mechanism for capturing and rendering real world materials. These datasets are expensive to collect and use too much memory to be practical for large scale computer graphics applications, such as full 3D VR environments. Current solutions leverage deep learning to compress these materials or rly on limited analytical solutions. Although these approaches improve usefulness in rendering pipelines, they fail to give artists deep control over the material appearance. This work presents a machine learning approach that generates a parameter space of the measured BRDF data to enable material editing. We explore applying β-Variational Autoencoders to disentangle the learned latent space. Demonstrations of these learned editable BRDFs are available at this link.

1. Introduction

Natural material appearance is complex with respect to many parameters. This large set of dimensions makes appearance hard to represent by a single analyitcal function. Bidirectional Reflection Distribution Function (BRDF) are the standard for representing materials. There are two kinds: analytical and measured. Analytical BRDFs have shown promise in describing various appearances [1, 2, 3]. However, analytical models are restricted in range and may diverge from real world properties. Measured BRDFs on the other hand exactly replicate real world properties at the cost of an understandable parameterization [4, 5, 6]. Existing methods that perform deep compression offer material interpolation and 2D latent space traversal for material editing. This pales in comparison to the power of analytical approaches.

Our contributions include:

- We propose a deep learning approach for a flexible parmaterization of measured BRDFs
- Develop a color-reflectance decomposition to improve human editability
- Design a friendly user-interface for editing and generating BRDFs in real time.

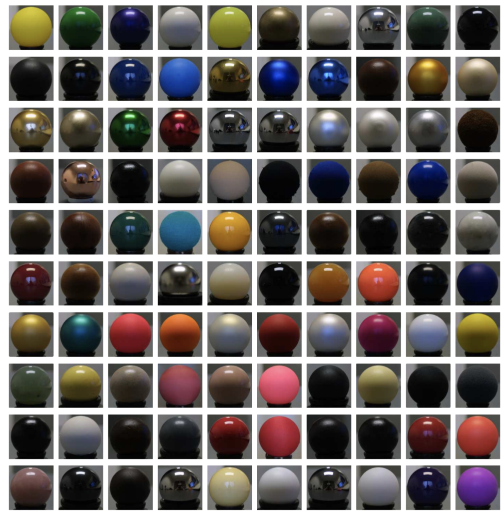

Figure 1. Examples of MERL's BRDFs

2. Related Work

2.1. Measured BRDFs

Mitsubishi Electric Research Laboratories (MERL) BRDF Dataset [7] Their acquisition system requires a spherically homogenous sample of the material. The system is placed in a completely isolated room painted in black matte. Then, a high dynamic range RGB image is taken at many camera viewing angles and light incident angles according to the sampling scheme. They choose Rusinkiewicz's coordinate system and discretize θ_h, θ_d into 90 bins, and ϕ_d into 180 bins. In total, each isotropic material is approximately 35MB and takes 4 hours to collect. Figure 1 demonstrates examples.

EPFL's Realistic Graphics Lab Material Database [8] This work introduces an adaptive parameterization to lower the required samples and improve resolution for specular materials. Each isotropic material on average takes under 1MB to store. Figure 2 demonstrates examples.

Figure 2. Examples of RGL's BRDFs

2.2. BRDF Editing

Analytical There have been many works that offer analytical models through mathematics and comparing accuracy to measured data. The primary challenge of this approach is to develop an analytical model complex enough to represent the richness of the measured data, yet simple enough so that the fitting process remains stable.

User studies Some works call for user studies to gather perceptual data such as how glossy a material is [9]. These have the advantage of producing a paramterization inline with human needs, but may fail at capturing the full complexity of real world materials.

Machine learning Machine learning approaches have been used to find manifold representations of measured BRDF data. In the original paper introducing the MERL BRDF dataset, a non linear dimension reduction technique was used [7]. By grouping similar materials on the manifold, they can find paths between those groups to edit the appearance. The approach to find the manifold and interpolation scheme has been refined in [10]. This process however is pseudo-random making it hard to control specific properties independently. Recent works use deep learning to compress the input measurements [11, 12, 5]. Their approaches offer some editing possibilities through linear interpolation.

3. Method

Our method trains a variational autoencoder to learn the distribution of measured BRDFs. At inference time, the encoder is discarded and users can change latent vector values to reconstruct new BRDFs. Our proposed network is visualized in Figure 3. We use two different training regimens Figure 4 and Figure 5. The first takes as input the entire BRDF cube ($3 \times \phi_d \times \theta_h \times \theta_d$) while the second ignore color and decomposes the cube into individual channels ($\phi_d \times \theta_h \times \theta_d$).

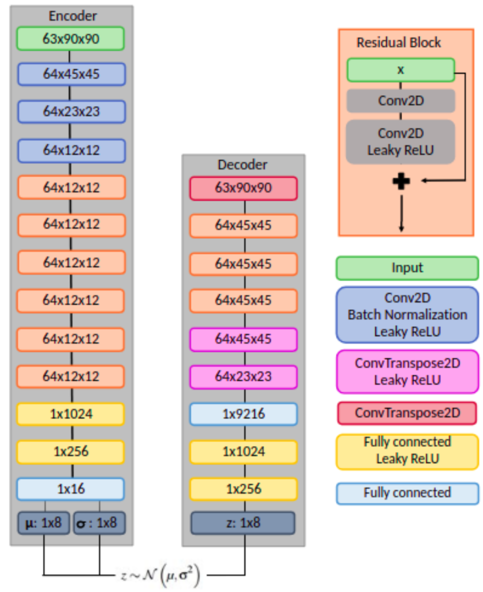

Figure 3. Network Architecture

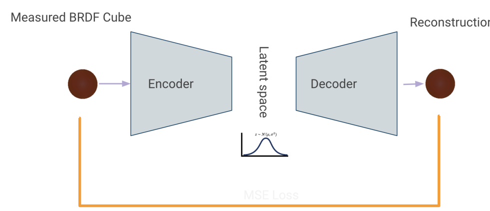

Figure 4. VAE Architecture

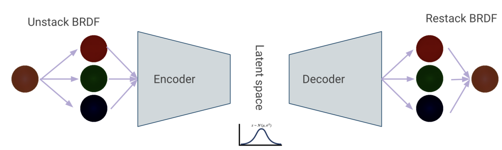

Figure 5. Colorless Architecture

3.1. Training Details

The network was implemented in PyTorch. All models were trained using the β-VAE system for 2000 epochs with a learning rate of 3×10^{-5}, and $\beta = 12$. We experimented with latent sizes from 3 to 16 but found model 1 performed best with 8 while model 2 with 3.

4. Experimental Results

4.1. Model 1 - MERL Trained

Our first model trained on just the MERL dataset demonstrated reasonable disengagement as seen in Figure 6. How-

Parameter	property
1	Diffuse color from blue to red
2	Sheen
3	Subsurface
4	Clear coat from blue to red
5	Specular to Diffuse
6	Haziness
7	Color lightness
8	Specular color from red to blue

Table 1. We associate a name to the factor controlled by each parameter through visual perception.

ever, we observed that the green color had poor reconstruction.

4.2. Model 2 - MERL+RGL Trained

Because the MERL dataset is biased against green (Figure 1), we decided to also mix in the RGL dataset and increase the latent size to 12. This resulted in better reconstructions, but worse disengagement.

4.3. Model 3 - New Parameters

We learned a relatively good disentangled parameterization for measured BRDFs in the MERL dataset from Model 1. Importantly, only 2 parameters control the color properties 1 and 8. In order to increase the color range, we introduce two new parameters that control diffuse and specular color from green to purple (Figure 7). We generate two materials, M_1 and M_2, from two sets of parameters. The first set uses parameters 1 through 8, the second set swaps 1 with 9 and 8 with 10. Finally, the materials are merged by replacing the green channel of M_1 with the red channel of M_2. This manual addition of parameters is only possible because the initial parameterization learned is interpretable and disentangled. Indeed, the disentanglement feature forces the color to be controlled by two parameters only and the interpretability allows us to modify the original purpose of the parameters. The two newly created parameters allow us to broaden the range of generated material colors.

4.4. Model 4 - Reflectance Only Model

A fundamental problem is that color cannot be captured via a single scalar unlike specularity and haziness. Instead, red, green and blue need to be mixed together to produce any color. Our model learned simple red-blue interpolation, but in order to solve the 3-point interpolation problem the model architecture needs to change. Because reflectance functions are wavelength dependent, in theory, each channel of the BRDF should be independent. Therefore, we can instead learn a general reflectance model and stack the color channels after the fact Figure 5. This produced the best reconstructions while maintaining some understandable parameters.

Method	MSE ↓
MERL Trained	0.0074
MERL+RGL Trained	0.0013
Reflectance Only	**0.0007**

Table 2. Reconstruction accuracy of the major methods presented.

4.5. Reconstructions

Using a simple lit-sphere shader based render in Python, we compute the mean squared error between the reconstructed BRDFs and the original BRDFs. Table 2 shows the error for each method. Observe that adding the RGL dataset improves reconstruction accuracy marginally, but separating color and reflectance shows the most improvement. Visually this is also obvious Figure 8.

4.6. UMAP Latent Traversal

Other neural compression works offer a 2D or 3D latent space traversal as a means of editing a BRDF. The latent vector z is mapped to a 2D embedding space using Uniform Manifold Approximationa nd Projection (UMAP) [13]. UMAP is particularly useful because the algorithm preserves the global data structure and is invertible. A 2D point can be dragged over the manifold and mapped back to a latent vector that is then passed through the decoder before rendering. We observe that this interface has limited editing control over generated material appearance Figure 9.

5. Editing Tools

We created 5 different editing experiences based on the ideas we presented. They are all web accessible via this link. An example of the interface is visualized in Figure 10. In addition, we developed an extension to Disney's BRDF Explorer to demonstrate editing with 2D latent traversals [1]. Informally, three non-author users played with each variation of the interface. After getting an internal sense of what each slider controls, they were tasked with achieving a certain edit, such as making a material shiner or bluer. User sentiment demonstrated interfaces with more precise control over color were preferable. We hypothesize this is because color changes are the fastest and most obvious to observe.

6. Course Items

In general we worked together on all items, but some specific individual contributions include:

Sachin

- Wrote/trained BRDF VAE
- Wrote/trained colorless VAE
- Developed base web-demo

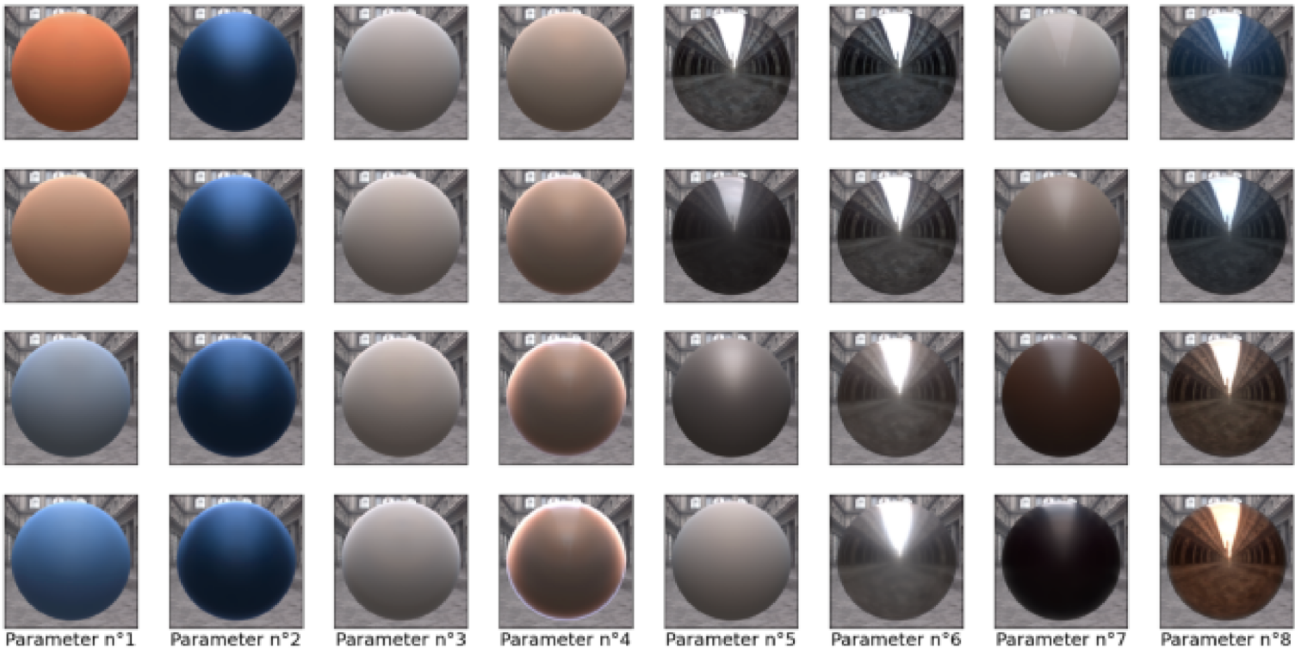

Figure 6. Latent traversal of each variable of the latent space. We observe that each variable controls a distinct generative parameter. Rough name assignments are shown in Table 1.

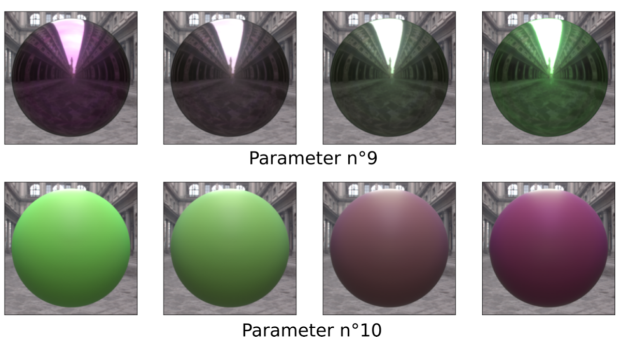

Figure 7. Example traversal of new parameters.

Sakshum

- Wrote MERL/RGL BRDF dataloader
- Wrote/trained TC-VAE
- Created 2D latent traversals

7. Future Work

Future work might include a more light-weight reflectance only decoder model to target even better deep compression. A user study with a range of professionals and average users should be conducted to determine if these new interfaces are better for material editing. Exploring other measured BRDF datasets that also include more ansiotropic materials would also be a useful direction. Separately, evaluating other editing schemes such as text and integration into neural radiance field factoring would be useful.

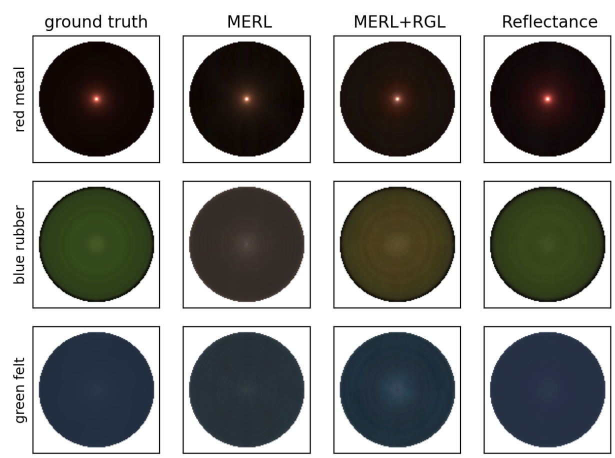

Figure 8. Selected reconstructions.

8. Conclusions

In an effort to combine the best of both worlds between the richness of measured BRDFs and the high control of analytical BRDF models, we developed a machine learning approach to create a disentangled space for measured BRDFs. Our method is self-supervised and has demonstrated strong visual interpretability. Futhermore, we ex-

Figure 9. Example of uniform sampling over the 2D latent space. Left, first model - MERL trained VAE model. Middle, second model - MERL+RGL trained VAE model. Right, third model - reflectance only VAE model. Observe that although many characteristics are represented, it is difficult to tweak any one independently.

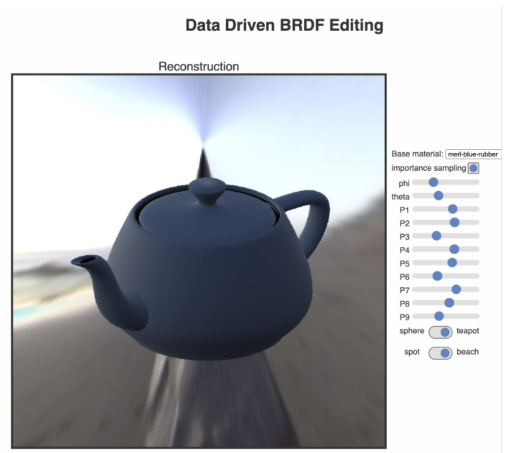

Figure 10. Example of web-based BRDF editor

plore color-reflectance decomposition for a better editing interface to overcome the 3-point interpolation problem.

References

[1] Burley B., Studios W. D. A.: Physically-based shading at disney. In ACM SIGGRAPH (2012), vol. 2012, vol. 2012, pp. 1–7. 1, 3

[2] Karis B., Games E.: Real shading in unreal engine 4. Proc. Physically Based Shading Theory Practice 4, 3 (2013), 1

[3] Colbert M., Pattanaik S., Krivanek J.: Brdf-shop: creating physically correct bidirectional reflectance distribution functions. IEEE Computer Graphics and Applications 26, 1 (2006) 1

[4] Chuankun Zheng, Ruzhang Zheng, Rui Wang, Shuang Zhao, and Hujun Bao. 2021. A Compact Representation of Measured BRDFs Using Neural Processes. ACM Trans. Graph. 41, 2, Article 14 (nov 2021), 15 pages. 1

[5] Bingyang Hu, Jie Guo, Yanjun Chen, Mengtian Li, and Yanwen Guo. 2020. DeepBRDF: A Deep Representation for Manipulating Measured BRDF. Computer Graphics Forum 39, 2 (2020), 157–166. 1, 2

[6] Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. 2021. Neural BRDF Representation and Importance Sampling. Computer Graphics Forum (2021). 1

[7] Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. ACM Transactions on Graphics 22(3), 759–769 (2003) 1, 2

[8] Jonathan Dupuy (Unity Technologies) and Wenzel Jakob (EPFL). An Adaptive Parameterization for Efficient Material Acquisition and Rendering. In ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018) 1

[9] Wills,J.,Agarwal,S.,Kriegman,D.,Belongie,S.:Toward a perceptual space for gloss. ACM Trans. Graph. 28(4) (2009) 2

[10] Soler, C., Subr, K., Nowrouzezahrai, D.: A Versatile Parameterization for Measured Material Manifolds. Com- puter Graphics Forum 37(2), 135–144 (2018) 2

[11] Hu, B., Guo, J., Chen, Y., Li, M., Guo, Y.: Deepbrdf: A deep representation for manipulating measured brdf. Computer Graphics Forum 39(2), 157–166 (2020) 2

[12] Zheng, C., Zheng, R., Wang, R., Zhao, S., Bao, H.: A compact representation of measured brdfs using neural processes. ACM Trans. Graph. 41(2) (2021) 2

[13] McInnes, L., Healy, J., Saul, N., Grossberger, L.: Umap: Uniform manifold approximation and pro jection. The Journal of Open Source Software 3(29), 861 (2018) 3