An Improved NeuMIP with Better Accuracy

An Improved NeuMIP with Better Accuracy

Bowen Xue[1] ,Shuang Zhao[2],Henrik Wann Jensen[3],Zahra Montazeri[1]
[1]University of Manchester, UK
[2]University of California, Irvine, USA
[3]Luxion, USA

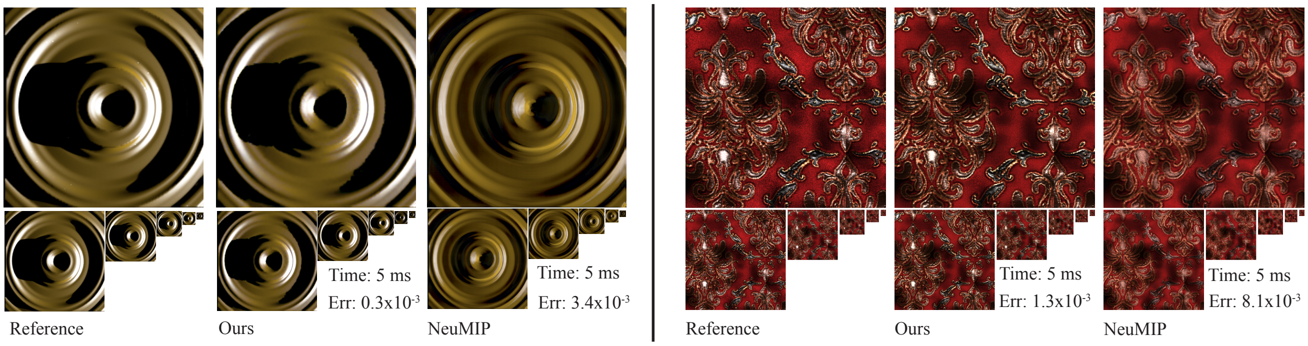

Figure 1: *(Left), comparison of our method to the baseline NeuMIP model and the reference, for different scales. The reference is based on displaced geometry, and by using networks of identical sizes, our neural models enjoy up to 90% lower MSE compared to NeuMIP. Our method can accurately capture the self-shadowing as well as sharp highlights of complex materials which are known as the limitations of the traditional NeuMIP. (Right), we show another material with a different lighting configuration to further demonstrate our accurate representation of high-frequency features.*

Abstract

Neural reflectance models are capable of accurately reproducing the spatially-varying appearance of many real-world materials at different scales. However, existing methods have difficulties handling highly glossy materials. To address this problem, we introduce a new neural reflectance model which, compared with existing methods, better preserves not only specular highlights but also fine-grained details. To this end, we enhance the neural network performance by encoding input data to frequency space, inspired by NeRF, to better preserve the details. Furthermore, we introduce a gradient-based loss and employ it in multiple stages, adaptive to the progress of the learning phase. Lastly, we utilize an optional extension to the decoder network using the Inception module for more accurate yet costly performance. We demonstrate the effectiveness of our method using a variety of synthetic and real examples.

CCS Concepts

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

1. Introduction

Modeling the appearance of real-world materials in a physically faithful fashion is crucial for predictive rendering. This, however, is a challenging task: Many materials exhibit comprise complex fine-grained geometries that largely drive their macro-scale appearances. Traditionally, material reflectance is typically specified using spatially varying BRDFs (SVBRDFs) or bidirectional texture functions (BTFs). While these models work adequately for many applications, they are typically limited to one specific scale (or resolution). Further, SVBRDFs have difficulties handling parallex effects while BTFs [DvGNK99] are highly data intensive.

Neural appearance modelings are a recent effort to address these issues utilizing the neural networks to learn the material reflectance representations [KMX*21; KWM*22; FWH*22]. Although some of these methods—such as NeuMIP [KMX*21]—can learn the appearance of any material at varying physical scales, they still have difficulties handling fine details and/or specular reflections.

Our goal is to extend these neural methods to better handle materials exhibiting complex light-transport effects—such as sharp highlights, detailed self-shadowing, and significant parallax effects—that previous neural methods have difficulties capturing.

In this work, we propose a novel neural representation that offers a new level of accuracy for modeling the appearances of materials with complex specular reflections. Concretely, we make the following contributions:

- We propose an improved extension to NeuMIP framework to better capture high glossy materials by introducing an input encoding step to the network to map the training inputs into a higher dimensional space (§3.1).
- For better robustness, we also introduce new losses to allow our model to better capture both high- and low-frequency effects (§3.2).

We demonstrate the effectiveness of our technique by comparing to the original NeuMIP [KMX*21] using several examples in Fig. 4, 6, and 5. In practice, similar to NeuMIP, our neural reflectance model can be integrated into most rasterization- and ray-tracing-based rendering systems.

2. Related Work

Neural rendering has emerged as a promising approach for a wide variety of applications, including material rendering, texture synthesis, and view synthesis. In this section, we review the most recent and relevant work in the area of neural rendering, focusing on techniques used for material rendering and displacement mapping.

Displacement mapping serves as a powerful technique for augmenting material complexity on surface geometries, thereby yielding persuasive parallax, silhouette, and shadowing effects. However, it imposes a considerable demand on computational resources. Conventional ray-tracing-based renderers typically effectuate displacement by tessellating the base geometry, a process that necessitates significant storage and computational capabilities [TBS*21].

A suite of techniques has been proposed as approximations to displacement mapping, including parallax mapping [KTI*01], relief mapping [OBM00; POC05], view-dependent displacement mapping (VDM) [WWT*03], and generalized displacement maps (GDM) [WTL*04; YZX*04]. These methodologies are fundamentally geometric and reliant on heightfields, overlooking the reflective effects emanating from the intricate material geometry.

bidirectional texture function (BTF) have been employed to represent arbitrary reflective surface appearances, first proposed by Dana et al. [DvGNK99]. The storage of a discretized 6D function incurs substantial costs, and a multitude of compression techniques have been scrutinized [HFM10]. Rainer et al. [RJGW19] introduced a neural architecture based on an autoencoder framework for compressing BTF slices per texel, and later advanced their work by integrating diverse materials into a shared latent space.

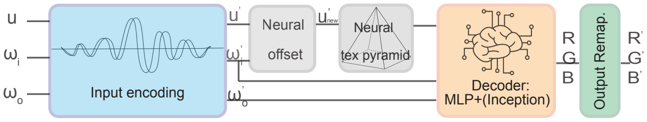

Figure 2: ***Overview of our enhanced neural architecture.***

NeuMIP [KMX*21; KWM*22], an innovative neural approach, has been formulated to render and represent materials across disparate scales efficiently. Despite its advantages, NeuMIP faces constraints due to its network architecture and design, struggling to simulate the high-frequency information inherent in materials. Furthermore, it fails to accommodate curved surfaces. A more recent endeavor [KWM*22] aimed to overcome these shortcomings by incorporating surface curvature and transparency information into the neural model. Yet, the task of capturing high-frequency materials remains a formidable challenge.

Recent advancements in Neural Radiance Fields (NeRF) have led to the development of methods capable of handling complex materials and geometries. The NeRF technique [MST*21], which represents scenes as continuous volumetric fields, has been adapted for a plethora of applications, such as NeRF-W [ZZF*22] to manage view-dependent appearances, and Fourier plenOctrees for NeRF [WZL*22] for real-time rendering. The methodology employed in this paper is inspired by NeRF to capture details with sparse sampling.

Volumetric Representations and Thin Shells have been harnessed in recent research to model materials as thin volumetric layers surrounding a base surface mesh. Jakob et al. [JAM*10] proposed a unified framework for the real-time rendering of intricate materials using a thin shell method. Zhao et al. [ZJMB11] introduced volumetric fabric models based on data sourced from micro-CT imaging, which was subsequently mapped onto a curved surface using shell mapping. Lombardi and his team [LSSS18] demonstrated a neural material approach utilizing volumetric layers, capable of rendering high-quality outputs, including precise silhouette and parallax effects.

Micro-geometry appearance models grapple with the granular details of the material and provide high-fidelity rendering results. The realistic rendering of fabrics, for instance, continues to be an elusive goal despite substantial efforts [KSZ*15; MXF*21]. More recently, Montazeri et al. [MGZJ20] introduced an efficient and unified shading model for woven and subsequently knit [MGJZ21] fabrics, though these models do not address multi-resolution. In this study, we exploit their model to generate our fabric samples for training data.

3. Our Method

In this section, we describe our neural method for representing and rendering challenging materials, that previous works have difficulty reproducing. We use NeuMIP [KMX*21] as a baseline and improve upon their model for a variety of materials. In what follows, we show our enhanced network design (§3.1) yields to better capture the high frequency. Then, we explain our optimization strate-

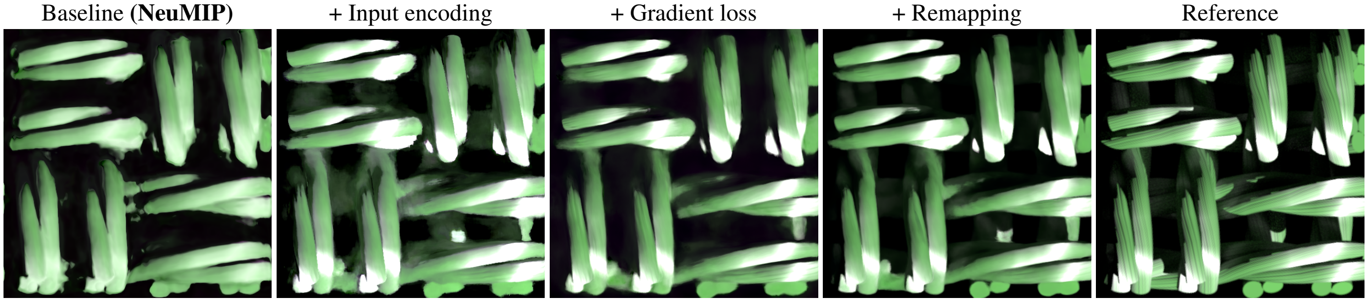

Figure 3: *We show the contributions of each of the improvemental steps we proposed to enhance the baseline, NeuMIP model. Our method uses the same decoder as NeuMIP and do not add any overhead, yet, is notably more accurate than the baseline.*

gies (§3.2) to further improve the performance of training the complex materials. Our results in comparison with baseline NeuMIP are listed in Fig. 4.

3.1. Enhanced Network Design

Overview of NeuMIP. The input to NeuMIP is a 7D parameter set including the position u, incoming and outgoing direction ω_i, and ω_o as well as the kernel size to address the material scale. Their pipeline consisted of three main stages, first to update the position u to compensate the micro-geometry using a neural offset module, then passed to the neural texture pyramid to handle different levels of detail, and lastly, an MLP decoder to generate the image. Our framework performs in a similar fashion to the classical NeuMIP, as illustrated in Fig. 2 with the following improvements, as an extended model. **Input decodings:** A main limitation of the NeuMIP pipeline is handling high-fidelity materials, especially at a small scale. Materials with sharp highlight cannot be captured using the MLP decoder network proposed in NeuMIP regardless of parameter tuning and loss function choices. We proposed a new architecture to address this shortcoming, drawing inspiration from the work on NeRF. The positional encoding techniques proposed in NeRF, play a crucial role in improving high-frequency details, however, it only accounts for changes in camera and 3D object positions.

In a similar fashion, we incorporate high-frequency encoding for lighting direction ω_i and camera directions ω_o, along with texture position u. Rahaman et al. [RBA*19] showed neural networks are biased toward learning low-frequency functions and perform poorly at representing high-frequency variation. Thus, we modify the MLP decoder by mapping its inputs to a high-dimensional space using Fourier transformation [BB86] instead of directly operating on input coordinates, such as in previous work.

This mapping significantly improves the ability of the network to reconstruct highlights and capture high-frequency image features, addressing the shortcomings of the traditional NeuMIP network. The formulation of our decoder F is a composition of two functions $F = F' \circ \gamma$. Where only F' is learned and $\gamma(.)$ is the encoding function that is applied to each of the input values which are all normalized, $p \in u, \omega_i, \omega_o$

$$\gamma(p) = \left(sin(2^0 \pi p), cos(2^0 \pi p), ..., sin(2^{L-1} \pi p), cos(2^{L-1} \pi p) \right),$$

where L defines the level of frequencies. Based on our experiments, we set L as 10 and 4 for $\gamma(u)$ and $\gamma(\omega)$, respectively. Fig. 3 shows a comparison with and without the input encoding stage for a sample scene to highlight its contribution.

Inception module (optional): In addition to the input encodings $\gamma(.)$, to further improve capturing the high frequencies, we propose an optional step for using an Inception module instead of MLP layers, used in classical NeuMIP. Introducing Inception blocks comes with the expense of slower training and evaluation time, due to the larger number of network weights. Hence, we refer to this as *optional* step and do not include it in our comparison results with NeuMIP for fairness. The contribution of our extended architecture capturing the fine details is exhibited in Fig. 8.

The Inception modules were first introduced in [SLJ*15] is a network block that optimizes the structure of a convolutional network based on the image model. It allows multiple types of filter sizes instead of a constant one, such as for fully-connected layers. This unique network design leveraging the Inception layers enables capturing the image features in multiple scales and reproducing details while preserving the overall appearance simultaneously. Introducing an adaptive kernel size into our network yields better capturing of image features in different scales.

3.2. Enhanced Loss Functions

While the two aforementioned contributions in the network design explained in §3.1, significantly improve the NeuMIP baseline, the fine details are not still fully reconstructed such as the fiber geometries shown in Fig. 3. To further improve learning the high-frequency features, we propose a series of optimization strategies to emphasize high frequencies and better reproduce fine details.

Gradient loss.: In order to better preserve edge and high-frequency details during the training process, we propose an innovative gradient-based loss function. The primary shortcomings in neural-network-generated results pertain to the absence of textural details, hence, we draw inspiration from the Canny edge detection algorithm [Can86] to keep textural details. Consequently, we employ the Sobel operator to extract edge information from the images as a preprocessing phase. We calculate the gradient of the predicted and reference images prior to computing the loss difference and we refer to this as the Gradient Loss $\mathcal{L}_G$.

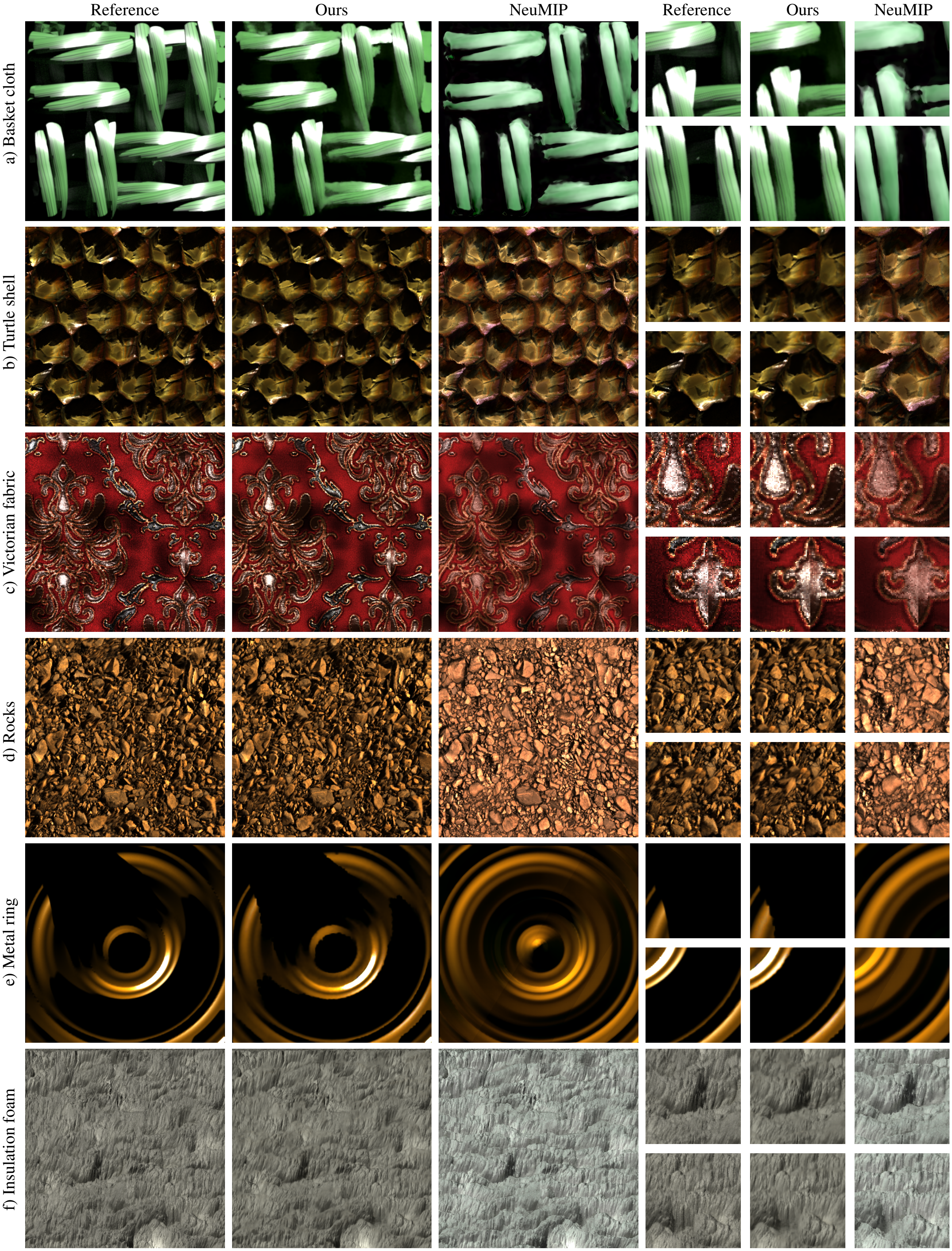

Figure 4: *Comparisons of synthetic data.*

Sobel and Feldman [SF73] introduced a 3x3 kernel as a discrete differential operator for capturing edges in both horizontal (x) and vertical (y) directions, as demonstrated below:

$$k_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{and} \quad k_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

G_x and G_y are computed by convolving the image I using the filters as $k_x * I$ and $k_y * I$, respectively. The gradient loss function used at backpropagation is as follows, where $\hat{G}$ is the reference image.

$$\mathcal{L}_G(I) = \left(\hat{G}_x(I) - G_x(I)\right)^2 + \left(\hat{G}_y(I) - G_y(I)\right)^2 \qquad (1)$$

Please refer to Fig. 3 to exclusively compare the absence of gradient-based loss as it notably improves the learning capability of the network capturing the details. We postulate that the proposed loss function exhibits a high degree of generalizability and can be extensively employed in various image generation tasks, such as super-resolution, novel view synthesis, and image-to-image translation.

Output remapping : While our gradient-based loss achieves a substantial improvement in handling the edges, it also introduces bias in those regions. To compensate for this, we utilize a non-linear image value remapping strategy to increase the contribution of low-luminance regions. After rigorous experiments, we noticed applying 4^{th} root functions as the image remapping is the optimum spot to capture both low and high frequencies. Our final loss function is formulated as follows:

$$\mathcal{L} = \frac{1}{n}\sum_{i=0}^{n}\left(\mathcal{L}_1(I) + \mathcal{L}_G(\frac{1}{I^4})\right), \qquad (2)$$

where $\mathcal{L}_G$ is formulated in (1), $\mathcal{L}_1$ is simply $\frac{1}{n}\|\hat{I} - I\|$, and n is the total pixel count.

4. Implementation Details

Dataset and training: The input to our neural rendering framework is 7D queries including the camera and light direction (ω_i, and ω_i), UV location u and the kernel size. Our method directly outputs the radiance value per texel and uses this as back-propagation during optimization. We generate our datasets (Basket cloth and Metal ring) using production renderer, Keyshot [Lux20] and the rest is published datasets by NeuMIP [KMX*21]. The Metal ring reference is explicit displaced geometry using heightfield, the same as NeuMIP datasets. The Basket cloth dataset, however, is constructed of actual micro-geometry as explicit curves then shaded using [MGZJ20], which makes it even more challenging. During training, the evaluation of the network happens for the whole batch and proceeds until convergence, which is $30,000$ for our model, and $80,000$ in case of an additional Inception module.

Rendering: Our extended NeuMIP can be integrated into a Monte-Carlo renderer, as such in NeuMIP, and capture the interaction bounces between objects in the scene. The results shown in this paper use an implementation in Mitsuba rendering engine [Jac20], accounting only for direct illumination. During render time, we use material query buffers to compute the inputs to our framework, then we pass the whole buffer to GPU to evaluate the queries as a batch. The LoD is also accounted in rendered results based on the camera distance per query. The rendered results in Fig. 7 has 16 sample-per-pixel (ssp) and the rest is 64 SPP.

4.1. Performance

A single evaluation of our method for a 512^2 image takes about 5ms which is the same as our main framework NeuMIP [KMX*21], as we use the same resources hence, on average, ours performs roughly the same time as NeuMIP, with significant quality improvements. Furthermore, the training of the results in this paper takes about 90 minutes for our main framework which is also the same as NeuMIP. The training and evaluation of our method are the same as NeuMIP underlying the same decoder. The measurements are performed using the maximum resolution 512^2 using NVIDIA V100 GPU.

5. Experiments and Results

In what follows, we show rendered results generated using our method. We first highlight our ablation studies to demonstrate the effect of the components of our proposed model (§5.1). Then, we evaluate the ability of our technique to represent a range of materials (§5.2).

5.1. Ablation Studies

In Fig. 3, we exhibit the importance of each of the components of our proposed method using the Basket cloth sample. This is a challenging material example for the neural method due to its complex geometry, substantial parallax effect, and high-frequency details such as fibers and sharp highlights. Using rigorous experiments, we observed our input encoding module and our gradient loss always improves capturing the edges and high-frequency features. While the inception module improves matching the overall

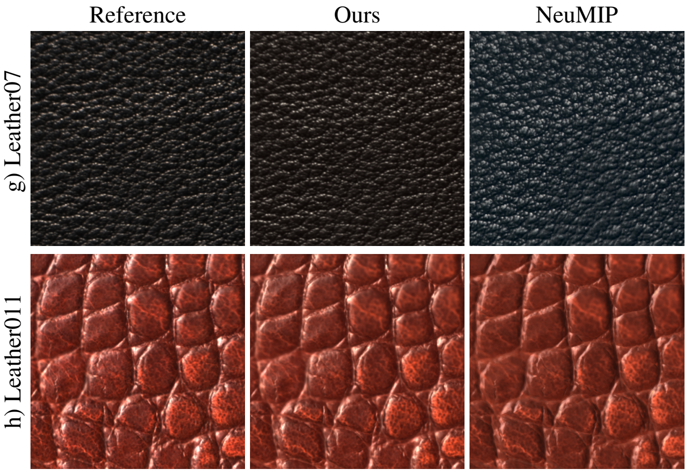

Figure 5: ***Comparisons of real BTF data.***

color better and the stage loss, enhances the bias between low and high-frequency features.

5.2. Evaluation Results

Comparisons with previous works: In Fig. 4, we show the results of our method on a wide range of complex materials, in comparison with traditional NeuMIP. In the Basket cloth sample, please note the deeper yarns are missing in NeuMIP while ours can successfully reproduce low-luminance regions as well as the high-frequency features such as edges and fiber details. In both the Turtle shell and Victorial fabric samples, NeuMIP has difficulty capturing the sharp highlights correctly. For the Metal ring scene, we noticed the shadow is not properly represented due to their simple MLP decoder. There is always a noticeable color shift in all complex datasets using NeuMIP, while less challenging materials work just as fine as our method.

Multi-resolution results: We demonstrate the effectiveness of our method addressing the different levels of detail of the material in Fig. 6. As expected, for the courser levels, the errors become smaller as we travel down in the hierarchical structure. This is due to the natural downsampling effect and gradual fading of the high-frequency details. We refer to the closest view as level-0 and the coarser levels are assigned to higher grades. The error score for different levels are highlighted in Table 2.

Real BTF results: The input to our neural method framework is a 7D parameter set that can be obtained using either synthetic datasets or real-measured BTF. We used examples from UBO 2014 dataset [MK14] to exemplify the effectiveness of our model regardless of the input source. The rendered results of BTF samples are shown in Fig. 5 for only a single level because the dataset does not include multi-resolution measurements. In comparison with NeuMIP as the baseline, our method performs better and matches the overall color more accurately.

Quantitative evaluation: We also measure the numerical error of our neural method when compared to the reference. Our method performance in comparison with NeuMIP is listed in Table 1 using both MSE loss (Means Square Error) and perceptual loss LPIPS (Learned Perceptual Image Patch Similarity). This shows the overall average on the whole dataset and our method always outperforms NeuMIP using the same configuration. Later, in Table 2 we demonstrate the scores of multiple scales of ours and NeuMIP model using the different levels of detail from the reference dataset. This is computed using the images in Fig. 6.

5.3. Our Extended Decoder

Our extended network structure is demonstrated in Fig. 9. The two 1*1 convolution layers at both ends, act as fully-connected and adjust the input and output size. The 6-layer Inception modules in between, capture the image features in four different scales using four kernel sizes that perform in parallel. Each of these Inception blocks consists of four parallel pathways. The first three pathways employ convolutional layers with window sizes of 1x1, 3x3, and 5x5, respectively, to extract information at different spatial scales. The middle two pathways first apply a 1x1 convolution to the input to reduce the number of input channels, thereby decreasing the model complexity. The fourth pathway utilizes a 3x3 max-pooling layer, followed by a 1x1 convolutional layer to alter the number of channels. All four pathways implement appropriate padding to ensure that input and output heights and widths remain consistent.

	MSE $x10^3$		LPIPS	
Scene ↓	Ours	NeuMIP	Ours	NeuMIP
a) Basket cloth	0.395	28.071	0.104	0.341
b) Turtle shell	0.0561	0.703	0.070	0.163
c) Victorian fabric	1.335	8.071	0.104	0.141
d) Rocks	0.096	0.384	0.0673	0.127
e) Metal ring	0.342	3.474	0.139	0.215
f) Insulation foam	4.133	41.436	0.049	0.243
g) Leather07	0.054	1.690	0.0325	0.342
h) Leather11	0.029	0.540	0.0125	0.134

Table 1: *Errors for images in Fig. 4 and 5.*

The output channel count for the Inception block is $64+128+32+32=256$, with the output channel ratios for the four pathways being $64:128:32:32=2:4:1:1$. Each Inception module is designed to have an input and output of 256 channels. In our comparative analysis of fully connected networks with equivalent depth and neuron count, we observed a markedly inferior performance in the fully connected networks. Interestingly, increasing the number of neurons and deeper fully connected networks does not yield significant improvements in performance in capturing the details.

6. Discussion and Conclusion

Limitation and future work: To integrate our neural reflectance models into physics-based Monte Carlo rendering frameworks, efficient importance sampling techniques for these models need to be developed—which we think is an important problem for future investigation.

Further, adopting our technique to improve the accuracy of the more recent neural reflectance model [KWM*22] (with better silhouettes) is worth exploring.

Lastly, generalizing our technique to introduce neural BSSRDFs (that can capture subsurface scattering) can be beneficial to many future applications.

Conclusion: In this paper, we improved the accuracy of the NeuMIP [KMX*21] by introducing a new neural representation as well as a training process for this representation. Using neural networks with identical sizes, compared with NeuMIP, our neural representation is capable of reproducing detailed specular highlights and shadowing at significantly higher accuracy while better preserving a material's overall color. Additionally, we proposed an optional modification to the decoder architecture that further enhances the performance. We demonstrated the effectiveness of our technique by comparing to NeuMIP (at equal network size) using several examples.

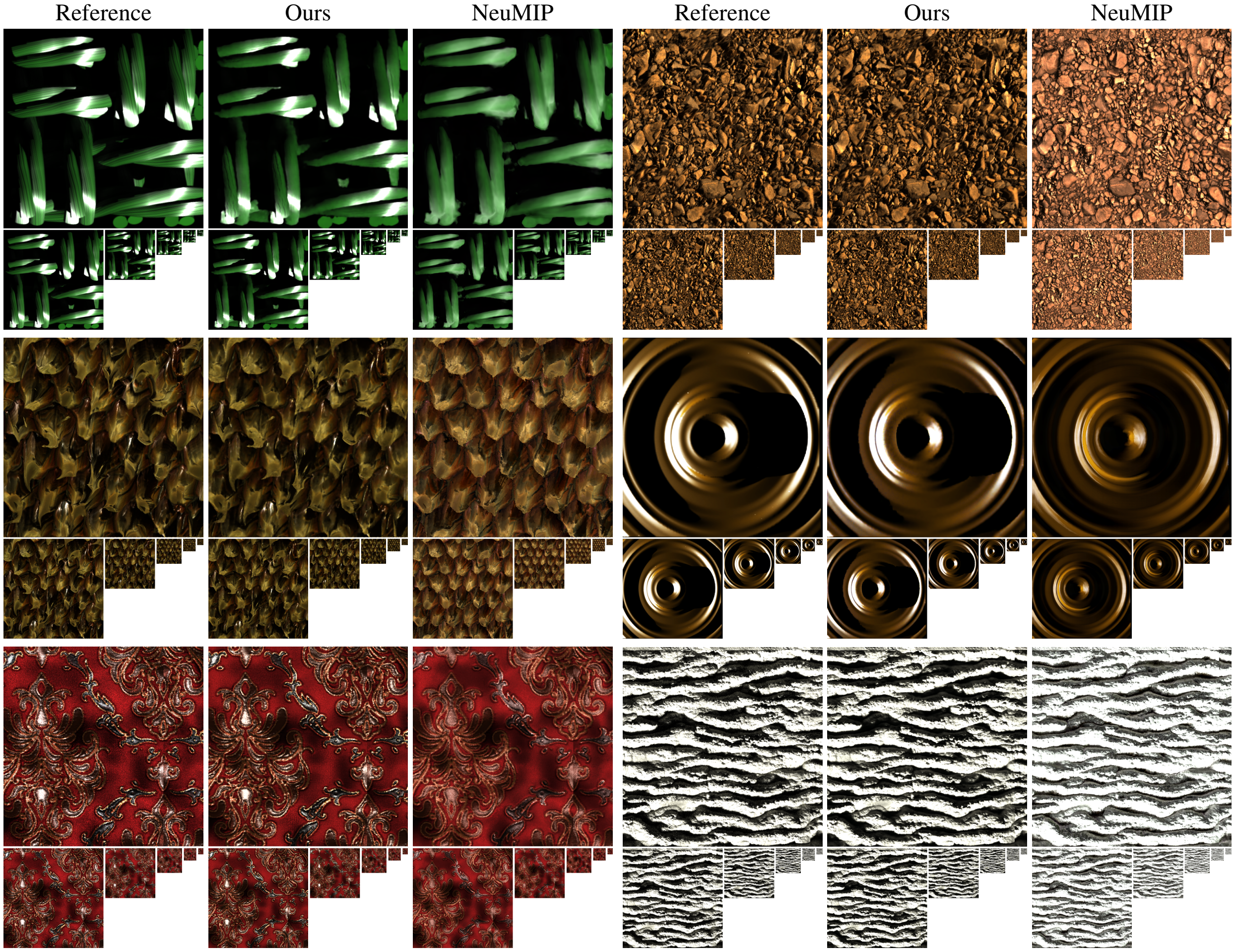

Figure 6: *Rendered results at the different levels of detail for selected materials.*

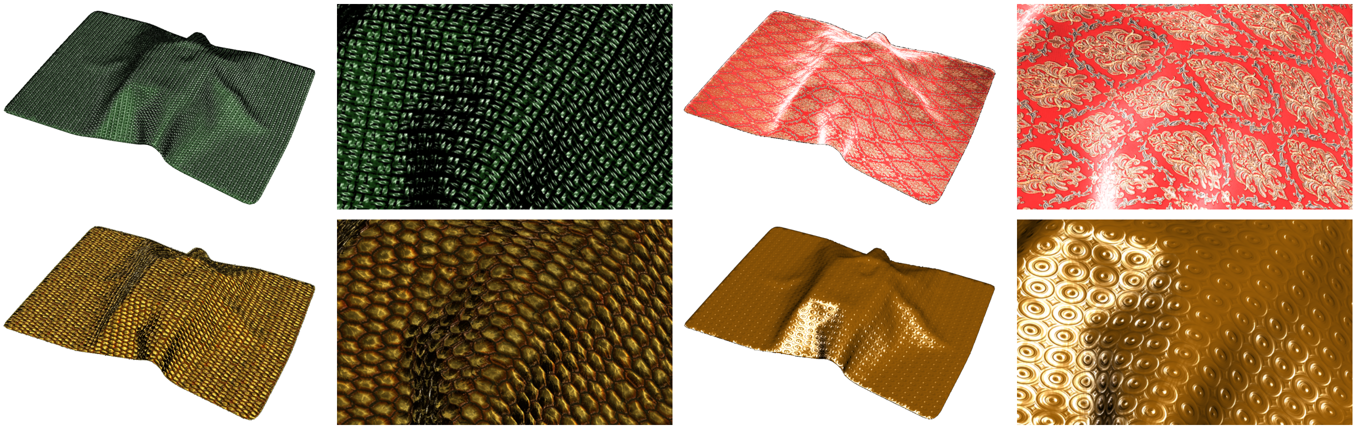

Figure 7: *An assortment of materials with our method on a non-flat surface. Please view the animation video in our supplemental materials. We animate moving light direction as well as gradual zoom to show consistency over time as well as different levels of detail, respectively.*

References

[BB86] Bracewell, Ronald Newbold and Bracewell, Ronald N. *The Fourier transform and its applications*. Vol. 31999. McGraw-Hill New York, 1986 3.

	Ours MSE $x10^3$						NeuMIP MSE $x10^3$					
Scene ↓ LoD →	0	1	2	3	4	5	0	1	2	3	4	5
a) Basket cloth	0.421	0.266	0.151	0.0753	0.0361	0.0175	9.651	9.180	7.462	6.453	6.204	4.324
b) Turtle shell	0.563	0.163	0.073	0.033	0.0145	0.0105	8.677	7.373	6.553	5.842	5.230	3.132
c) Victorian fabric	0.356	0.035	0.013	0.00681	0.00435	0.00215	0.672	0.167	0.0952	0.0662	0.0463	0.0254
d) Rocks	4.815	3.399	2.904	2.635	2.506	1.324	4.935	3.272	2.771	2.507	2.383	2.102
e) Metal ring	0.610	0.491	0.383	0.264	0.161	0.085	7.478	7.137	6.493	5.333	3.933	2.765
f) Insulation foam	0.137	0.0394	0.0175	0.00784	0.00384	0.00287	1.324	1.128	1.056	1.017	0.997	0.853

Table 2: *Errors for images rendered across multiple levels of details as shown in Fig. 6.*

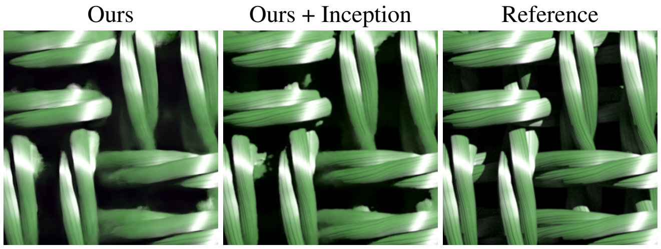

Figure 8: *Our optional decoder using Inception modules brings up the fine details, such as fiber curves in this cloth example. This makes the network larger, and unfair to evaluate with the baseline, hence we exclude this from "Ours" in the comparison results.*

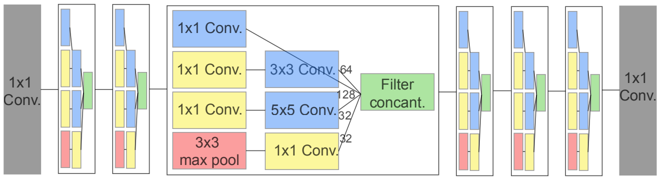

Figure 9: *We use four layers of MLP for our decoder, same as NeuMIP. However, for additional accuracy, we extend the decoder as an optional step for better performance. It contains six inception modules wrapped with two fully-connected layers at both ends.*

[Can86] CANNY, JOHN. "A computational approach to edge detection". *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), 679–698 3.

[DvGNK99] DANA, KRISTIN J., van GINNEKEN, BRAM, NAYAR, SHREE K., and KOENDERINK, JAN J. "Reflectance and Texture of Real-World Surfaces". *ACM Trans. Graph.* 18.1 (Jan. 1999), 1–34. ISSN: 0730-0301. DOI: 10.1145/300776.300778. URL: https://doi.org/10.1145/300776.300778 1, 2.

[FWH*22] FAN, JIAHUI, WANG, BEIBEI, HASAN, MILOS, et al. "Neural Layered BRDFs". *ACM SIGGRAPH 2022 Conference Proceedings*. SIGGRAPH '22. Vancouver, BC, Canada: Association for Computing Machinery, 2022. ISBN: 9781450393379. DOI: 10.1145/3528233.3530732. URL: https://doi.org/10.1145/3528233.3530732 1.

[HFM10] HAVRAN, V., FILIP, J., and MYSZKOWSKI, K. "Bidirectional Texture Function Compression Based on Multi-Level Vector Quantization". *Computer Graphics Forum* (2010). ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2009.01585.x 2.

[Jac20] JACOB, WENZEL. *Mitsuba*. https://www.mitsuba-renderer.org. 2020 5.

[JAM*10] JAKOB, WENZEL, ARBREE, ADAM, MOON, JONATHAN T, et al. "A radiative transfer framework for rendering materials with anisotropic structure". *ACM SIGGRAPH 2010 papers*. 2010, 1–13 2.

[KMX*21] KUZNETSOV, ALEXANDR, MULLIA, KRISHNA, XU, ZEXIANG, et al. "NeuMIP: Multi-Resolution Neural Materials". *Transactions on Graphics (Proceedings of SIGGRAPH)* 40.4 (July 2021) 1, 2, 5, 6.

[KSZ*15] KHUNGURN, PRAMOOK, SCHROEDER, DANIEL, ZHAO, SHUANG, et al. "Matching Real Fabrics with Micro-Appearance Models." *ACM Trans. Graph.* 35.1 (2015), 1–1 2.

[KTI*01] KANEKO, TOMOMICHI, TAKAHEI, TOSHIYUKI, INAMI, MASAHIKO, et al. "Detailed shape representation with parallax mapping". *Proceedings of ICAT*. Vol. 2001. 2001, 205–208 2.

[KWM*22] KUZNETSOV, ALEXANDR, WANG, XUEZHENG, MULLIA, KRISHNA, et al. "Rendering Neural Materials on Curved Surfaces". *SIGGRAPH '22 Conference Proceedings* (2022). DOI: 10.1145/3528233.3530721 1, 2, 6.

[LSSS18] LOMBARDI, STEPHEN, SARAGIH, JASON, SIMON, TOMAS, and SHEIKH, YASER. "Deep appearance models for face rendering". *ACM Transactions on Graphics (ToG)* 37.4 (2018), 1–13 2.

[Lux20] LUXION. *KeyShot*. https://www.keyshot.com. 2020 5.

[MGJZ21] MONTAZERI, ZAHRA, GAMMELMARK, SOREN, JENSEN, HENRIK W, and ZHAO, SHUANG. "A Practical Ply-Based Appearance Modeling for Knitted Fabrics". *arXiv preprint arXiv:2105.02475* (2021) 2.

[MGZJ20] MONTAZERI, ZAHRA, GAMMELMARK, SØREN B, ZHAO, SHUANG, and JENSEN, HENRIK WANN. "A practical ply-based appearance model of woven fabrics". *ACM Transactions on Graphics (TOG)* 39.6 (2020), 1–13 2, 5.

[MK14] MICHAEL WEINMANN, JUERGEN GALL and KLEIN., REINHARD. "Material Classification based on Training Data Synthesized Using a BTF Database". *ECCV* (2014) 6.

[MST*21] MILDENHALL, BEN, SRINIVASAN, PRATUL P, TANCIK, MATTHEW, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". *Communications of the ACM* 65.1 (2021), 99–106 2.

[MXF*21] MONTAZERI, ZAHRA, XIAO, CHANG, FEI, YUN, et al. "Mechanics-Aware Modeling of Cloth Appearance". English. *IEEE Transactions on Visualization and Computer Graphics* (Jan. 2021), 137–150. ISSN: 1077-2626. DOI: 10.1109/TVCG.2019.2937301 2.

[OBM00] OLIVEIRA, MANUEL M, BISHOP, GARY, and MCALLISTER, DAVID. "Relief texture mapping". *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, 359–368 2.

[POC05] POLICARPO, FÁBIO, OLIVEIRA, MANUEL M, and COMBA, JOAO LD. "Real-time relief mapping on arbitrary polygonal surfaces". *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. 2005, 155–162 2.

[RBA*19] Rahaman, Nasim, Baratin, Aristide, Arpit, Devansh, et al. *On the Spectral Bias of Neural Networks*. 2019. URL: https://openreview.net/forum?id=r1gR2sC9FX 3.

[RJGW19] Rainer, Gilles, Jakob, Wenzel, Ghosh, Abhijeet, and Weyrich, Tim. "Neural BTF compression and interpolation". *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, 235–244 2.

[SF73] Sobel, I. and Feldman, G. "A 3x3 Isotropic Gradient Operator for Image Processing". *Pattern Classification and Scene Analysis* (1973), pp. 271–272 5.

[SLJ*15] Szegedy, Christian, Liu, Wei, Jia, Yangqing, et al. "Going deeper with convolutions". *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, 1–9. DOI: 10.1109/CVPR.2015.7298594 3.

[TBS*21] Thonat, Theo, Beaune, Francois, Sun, Xin, et al. "Tessellation-Free Displacement Mapping for Ray Tracing". *ACM Trans. Graph.* 40.6 (Dec. 2021). ISSN: 0730-0301. DOI: 10.1145/3478513.3480535. URL: https://doi.org/10.1145/3478513.3480535 2.

[WTL*04] Wang, Xi, Tong, Xin, Lin, Stephen, et al. "Generalized displacement maps". *Eurographics Symposium on Rendering (EGSR)*. 2004, 227–233 2.

[WWT*03] Wang, Lifeng, Wang, Xi, Tong, Xin, et al. "View-dependent displacement mapping". *ACM Transactions on graphics (TOG)* 22.3 (2003), 334–339 2.

[WZL*22] Wang, Liao, Zhang, Jiakai, Liu, Xinhang, et al. "Fourier plenoctrees for dynamic radiance field rendering in real-time". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 13524–13534 2.

[YZX*04] Yu, Yizhou, Zhou, Kun, Xu, Dong, et al. "Mesh editing with poisson-based gradient field manipulation". *ACM SIGGRAPH 2004 Papers*. 2004, 644–651 2.

[ZJMB11] Zhao, Shuang, Jakob, Wenzel, Marschner, Steve, and Bala, Kavita. "Building volumetric appearance models of fabric using micro CT imaging". *ACM Transactions on Graphics (TOG)* 30.4 (2011), 1–10 2.

[ZZF*22] Zhang, Jian, Zhang, Yuanqing, Fu, Huan, et al. "Ray priors through reprojection: Improving neural radiance fields for novel view extrapolation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 18376–18386 2.