



Universidad
Rey Juan Carlos

NEURAL NETWORKS FOR DIGITAL MATERIALS AND RADIANCE ENCODING

**Tesis
doctoral**

Premio Extraordinario



Carlos Miguel Rodríguez-Pardo Jiménez



This monograph is a publishing project by Academia Abierta



Collection
Doctoral thesis

Neural networks for digital materials and radiance encoding

Autor

Carlos Miguel Rodríguez-Pardo Jiménez

Supervisor

Elena Garcés García

Doctoral Program in Information and Communication Technologies.

International Doctoral School

2023

This monograph has been published in collaboration with the Office for Open Knowledge and Free Culture and the International Doctoral School for having received the extraordinary award at Rey Juan Carlos University.

How to cite

Rodríguez-Pardo Jiménez, C. M. (2023). *Neural networks for digital materials and radiance encoding*. Editorial Academia Abierta. Retrieved from <https://monografias.urjc.es/index.php/omp-urjc/catalog/book/43>

Deposit

URJC open archive (digital BURJC)

DOI: <https://doi.org/10.33732/TD-43>

© 2023. Carlos Miguel Rodríguez-Pardo Jiménez

© Of the texts: Carlos Miguel Rodríguez-Pardo Jiménez

© Of the images: their authors

© Cover image: JSCreative-LabAI_Arts, <https://pixabay.com/es/illustrations/ai-generado-cerebro-ondas-cerebrales-8620473/>

Editorial

Academia Abierta. Servicio de publicaciones de la URJC

C/Tulipán s/n

28933 Móstoles

<https://monografias.urjc.es/index.php/omp-urjc>

Interior design and cover

Glaux Publicaciones Académicas, SLU

Editorial contact

Laura de la Cruz / Tomás Zarza

servicio.publicaciones@urjc.es

Some rights reserved



This document is distributed under the Creative Commons Attribution-ShareAlike 4.0 International licence, available at <https://creativecommons.org/licenses/by-sa-4.0/>

*That he not busy being born
Is busy dying.*

— **Bob Dylan**
It's Alright, Ma (I'm Only Bleeding)

ABSTRACT

Realistic virtual scenes are becoming increasingly prevalent in our society, with a wide range of applications in areas such as manufacturing, architecture, fashion design, and entertainment, including movies, video games, and augmented and virtual reality. Generating realistic images of such scenes requires highly accurate illumination, geometry, and material models, which can be time-consuming and challenging to obtain. Traditionally, such models have often been created manually by skilled artists, but this process can be prohibitively time-consuming and costly. Alternatively, real-world examples can be captured, but this approach presents additional challenges in terms of accuracy and scalability. Moreover, while realism and accuracy are crucial in such processes, rendering efficiency is also a key requirement, so that lifelike images can be generated with the speed required in many real-world applications. One of the most significant challenges in this regard is the acquisition and representation of materials, which are a critical component of our visual world and, by extension, of virtual representations of it. However, existing approaches for material acquisition and representation are limited in terms of efficiency and accuracy, which limits their real-world impact. To address these challenges, data-driven approaches that leverage machine learning may provide viable solutions. Nevertheless, designing and training machine learning models that meet all these competing requirements remains a challenging task, requiring careful consideration of trade-offs between quality and efficiency.

In this thesis, we propose novel learning-based solutions to address several key challenges in physically-based rendering and material digitization. Our approach leverages various forms of neural networks to introduce innovative algorithms for radiance encoding, digital material generation, edition, and estimation. First, we present a visual attribute transfer framework for digital materials that can effectively generalize to new illumination conditions and geometric distortions. We showcase a use-case of this method for high-resolution material acquisition using a custom device. Additionally, we propose a generative model capable of synthesizing tileable textures from a single input image, which helps improve the quality of material rendering. Building upon recent work in neural fields, we also introduce a material representation that accurately encodes material reflectance while offering powerful editing and propagation capabilities. In addition to reflectance, we present a novel method for global illumination encoding that leverages carefully designed generative models to achieve significantly faster sampling than previous work. Finally, we propose two innovative methods for low-cost material digitization. With flatbed scanners as our capture device, we present a generative model that can provide high-resolution material reflectance estimations using a single image as input, while introducing an uncertainty quantification algorithm that increases its reliability and efficiency. Additionally, we present a novel method for digitizing fabric mechanical properties using depth images as input, which we extend with a perceptually-validated drape similarity metric. Overall, the contributions of this thesis represent significant advances in the fields of radiance encoding and digital material acquisition and edition, enhancing the quality, scalability, and efficiency of physically-based rendering pipelines.

ACKNOWLEDGEMENTS

This thesis is the result of several years of relentless hard work. However, none of this would have been possible without the support of an outstanding group of people, to whom I owe a tremendous debt of gratitude. First and foremost, to my advisor Elena Garcés, for the opportunity to embark on this journey, for everything I have learned from her, for making sure I had all the resources I needed to get my work done, and for the countless and tireless hours she spent helping me improve as a researcher. I would also like to express my heartfelt appreciation to Dan Casas and Jorge Lopez-Moreno, who have been incredible mentors, always ready to offer guidance and support, and from whom I have learned so much over the years.

I want to express my sincere gratitude to the team at SEDDI for their unwavering support over the years. I feel specially grateful for the opportunity to work closely with David Pascual, Sergio Ruiz, and Sofia Domínguez, who not only helped me professionally but also made our time together more fun and enjoyable (with great *padel* matches and even better inside jokes, and silly names for our AI model iterations). They have been the best teammates I could have ever asked for. In no particular order, I would also like to thank Henar Domínguez, Sergio Suja, Javier Fabre, Loreto Pérez, Daniel Rodríguez, and Luis Romero, who have also made this journey much more fun and have positively contributed in many ways to my work.

I would like to express my deepest appreciation to SEDDI as a company, which I joined shortly after it was founded. Their financial and moral support

has been crucial to the successful completion of this thesis, and I am grateful for their unwavering assistance throughout this journey. Thanks to their generosity, I was able to access all the resources I needed, and they allowed me to publicly share my work through publications. I believe that SEDDI's future is bright, and I am excited to witness their future accomplishments.

I would also like to thank my friends and family, for their support and patience over the past years; my students, for their trust; and the people at the MSLab at URJC and the GIAA at UC3M who have helped me throughout these years.

I want to acknowledge the invaluable contributions of the reviewers who have selflessly dedicated their time and effort to provide constructive feedback on our papers. Their rigorous reviews have been instrumental in improving the quality of our research. Moreover, I am deeply grateful to the open-source community and the authors of countless open-access papers that have inspired and shaped many of the ideas we have developed in this work. I firmly believe that open communication and collaboration are critical for advancing the fields of machine learning and computer vision. Thus, I sincerely hope that the research communities continue to publish their work, preferably with open-source implementations and data, to facilitate knowledge-sharing and foster innovation.

Last but not least, I am grateful for the public investment in research and innovation, as well as the Spanish public education system, without which this thesis would not have been possible. Looking forward, I sincerely hope that as a society, we can redouble our efforts to defend our public democratic institutions, which are crucial for building a fair, free, sustainable, and equitable world. It is imperative that we prioritize investing in public education and research to foster innovation, create new opportunities, and address the complex challenges we face today. By doing so, we can build a brighter future for ourselves and future generations.

Contents

Abstract	ix
Acknowledgements	xi
1 Introduction	1
• 1.1 Theoretical Background	3
• 1.1.1 Physically-Based Rendering	3
• 1.1.2 Geometry	5
• 1.1.3 Materials	8
• 1.1.4 Illumination	12
• 1.2 Open Problems	14
• 1.3 Goals & Contributions	19
• 1.3.1 Contributions	20
• 1.4 Measurable Outcomes	23
• 1.4.1 Publications & Patents	23
• 1.4.2 Awards & Other Merits	24
• 1.4.3 Student Supervision	25
• 1.4.4 Teaching, Talks, and Seminars	25
• 1.4.5 Reviewing Activity	26
• 1.4.6 Industrial Research Projects and Commercial Products	26
2 Visual Attribute Transfer with Photometry-guided Neural Networks	27
• 2.1 Introduction	28
• 2.2 Related Work	30
• 2.2.1 Visual Attribute Transfer	30
• 2.2.2 Textured Materials	31
• 2.2.3 SVBRDF Estimation	32
• 2.3 Problem Formulation	33
• 2.4 Learning Framework	34
• 2.4.1 Patch-based Training	34
• 2.4.2 Network Design	35
• 2.4.3 Implementation details	37
• 2.5 Dataset and Metrics	38
• 2.5.1 Dual-Resolution Captured Data	38
• 2.5.2 Attribute-specific Metrics for Evaluation	39
• 2.6 Evaluation	39
• 2.6.1 Invariance to Input Illuminations and Distortions	39
• 2.6.2 Equivariance to Affine Transforms	41

• 2.7 Results and Comparisons	43
• 2.7.1 Interactive Stylizations	43
• 2.7.2 Creation of Large Scale Digital Material Assets	45
• 2.7.3 Generalization to Similar Materials	49
• 2.7.4 Limitations	49
• 2.8 Conclusions and Future Work	49
• 2.A Additional Implementation Details	52
• 2.B Additional Comparisons	54
• 2.C Extension to SVBSDF Propagation	60
• 2.C.1 Mesoscale Propagation	61
• 2.C.2 Implementation Details	63
• 2.C.3 Results	66
3 A Single Image Generative Model for Tileable Texture Synthesis	69
• 3.1 Introduction	70
• 3.2 Related work	71
• 3.2.1 Texture Synthesis	71
• 3.2.2 Texture Tileability	73
• 3.2.3 Deep Internal Learning	73
• 3.3 Overview	74
• 3.4 Self-Supervised Texture Synthesis with Adversarial Expansion	75
• 3.4.1 Network Architecture	76
• 3.5 Tileable Texture Stack Sampling	78
• 3.5.1 Latent Space Tiling	78
• 3.5.2 Discriminator-guided Sampling	79
• 3.6 Implementation Details	81
• 3.7 Experiments	83
• 3.7.1 Generator Q Design	83
• 3.7.2 Inter-map Consistency	84
• 3.7.3 Loss Function Ablation Study	85
• 3.8 Results and Comparisons	86
• 3.9 Conclusions and Future Work	89
• 3.A Additional Details	92
• 3.B Additional Comparisons, Results and Ablations	93
4 Neural Fields for BTF Encoding and Transfer	99
• 4.1 Introduction	100
• 4.2 Related Work	101
• 4.3 Method	104

• 4.3.1 Inference	104
• 4.3.2 Training	105
• 4.4 Implementation details	107
• 4.5 Evaluation	109
• 4.5.1 Qualitative Analysis	109
• 4.5.2 Compression comparisons with previous work	111
• 4.5.3 Limitations	111
• 4.6 Applications	112
• 4.6.1 Reflectance Propagation and Tileable BTFs	112
• 4.6.2 Structural Material Edition	113
• 4.6.3 Multi Resolution Neural Materials	114
• 4.7 Conclusions	114
• 4.A Model Implementation Details	118
5 Uncertainty-Aware Neural Networks for High Resolution Digitization	121
• 5.1 Introduction	123
• 5.2 Related Work	125
• 5.2.1 Lightweight Material Capture	125
• 5.2.2 Uncertainty Quantification in Deep Learning	126
• 5.3 Method	127
• 5.3.1 Network Design	127
• 5.3.2 Loss Function	128
• 5.3.3 Data Augmentation	129
• 5.4 Uncertainty Quantification	129
• 5.5 Experimental Results	131
• 5.5.1 Dataset	131
• 5.5.2 Metrics	131
• 5.6 Evaluation	132
• 5.6.1 Ablation Study	132
• 5.6.2 Qualitative Analysis	132
• 5.6.3 Uncertainty Evaluation	135
• 5.6.4 Active Learning	135
• 5.6.5 Comparisons with Previous Work	138
• 5.6.6 Limitations	140
• 5.7 Conclusions	140
• 5.A Additional Implementation Details	140
• 5.A.1 Model Design	140
• 5.A.2 Model Training	141
• 5.A.3 Artifact Detection	142

• 5.A.4 Capture Details	143
• 5.A.5 Comparisons with Previous Work	143
• 5.B Dataset Analysis	146
• 5.C Additional Details And Results	148
6 Casual Capture of Fabric Mechanics	155
• 6.1 Introduction	156
• 6.2 Related Work	158
• 6.2.1 Parameter Estimation Methods	158
• 6.2.2 Pre-Trained Models and Transfer Learning	160
• 6.2.3 Similarity Metrics	161
• 6.3 Overview	161
• 6.4 Datasets	162
• 6.4.1 Data Augmentation	164
• 6.5 Fabric Mechanics from Depth Images	165
• 6.5.1 Neural Network Architecture	165
• 6.5.2 Quantitative Evaluation	167
• 6.6 A Similarity Metric for Drape	170
• 6.6.1 Image-based Similarity of Drape	170
• 6.7 Evaluation	172
• 6.7.1 Human Judgment Perceptual Similarity of Drape	173
• 6.7.2 Image-based vs. Human Perceptual Similarity	173
• 6.7.3 Qualitative Results	175
• 6.8 Conclusions	175
• 6.A Additional Implementation Details	178
• 6.B Additional Results	179
• 6.C Ablation Study: Similarity Metric Parameters	182
7 Neural Models for Global Illumination	187
• 7.1 Introduction	188
• 7.2 Related Work	189
• 7.2.1 Neural Sampling and Representations	190
• 7.3 Overview	192
• 7.3.1 Map Pre-processing	193
• 7.3.2 Computing Ground Truth Tabular PDFs	193
• 7.4 Learned Sampling and PDF evaluation	194
• 7.4.1 Formalization	194
• 7.4.2 Design of F	195
• 7.4.3 Training F	196

• 7.5 Environment Map Compression	197
• 7.6 Implementation Details	197
• 7.7 Dataset	199
• 7.8 Evaluation	199
• 7.8.1 PDF Fit Accuracy	200
• 7.8.2 Environment Map Reconstruction	202
• 7.8.3 Computational Cost	204
• 7.8.4 Rendering Comparisons	205
• 7.9 Conclusions	206
8 Conclusions	209
Bibliography	219
A Resumen	269
• A.1 Antecedentes	270
• A.2 Objetivos	274
• A.3 Metodologia	275
• A.4 Resultados	280
• A.5 Conclusiones	283

Figures

• 1.1 Renders of virtual scenes for different use cases	1
• 1.2 Images of different real-world materials and textures	2
• 1.3 Example of light transport and material interactions	4
• 1.4 Example of light transport, connecting light sources to pixels	6
• 1.5 2D depiction of a physically-principled BSDF theoretical model	7
• 1.6 Example of spatially varying illumination encoding with Spherical Harmonics.	12
• 1.7 Schemes and photos of SEDDI's optical device	15
• 1.8 Example of a non-tileable and a tileable material	16
• 1.9 Renders of real and simulated garments	18
• 1.10 Overview of the structure of this thesis	19
• 1.11 Lightweight material digitization enabled by the methods presented in this thesis.	22
• 2.1 Illustration of the capabilities of our attribute transfer framework	27
• 2.2 Teaser Figure for photometricNet.	28
• 2.3 Overview of photometricNet.	33
• 2.4 Illustration of our proposed color augmentation policy	34
• 2.5 An overview of our evaluation dataset for PhotometricNet.	36
• 2.6 Qualitative results of our models under different datasets and data augmentation configurations	38
• 2.7 Ablation study on the influence of the size of the photometric dataset on the quality of the propagations	41
• 2.8 Output of our model under different distorted inputs	42
• 2.9 Quantitative evaluation of the robustness of our networks with respect to different transformations	43
• 2.10 Comparisons of our method with previous work on stylizations	44
• 2.11 Comparisons of our method with previous work on image analogies	46
• 2.12 Results of our framework for interactive material-aware visual attribute transfer.	46
• 2.13 Results of our framework for material capture using a smartphone	47
• 2.14 Qualitative comparison of photometricNet with previous work on guided attribute transfer	47
• 2.15 Results of our method on real BTF captured data	48
• 2.16 Generalization capabilities of photometricNet when evaluated on materials similar to those in their training dataset	50
• 2.17 Failure cases of photometricNet	51

• 2.18 An overview of our deep learning architecture	53
• 2.19 Impact of the depth of the model on its accuracy	53
• 2.20 Impact of the choice of the loss function used to train the networks in the estimated maps at macroscale	55
• 2.21 Relative position of the 27 directional light sources used to capture the datasets to train photometricNet	56
• 2.22 Additional comparison of photometricNet with related image analogies methods.	58
• 2.23 Additional comparisons of photometricNet with related image stylization methods.	59
• 2.24 A comparison of photometricNet with material digitization methods	60
• 2.25 Renders of SVBSDF maps transferred by our method	60
• 2.26 Virtual scene rendered using many materials generated by our method	61
• 2.27 Comparison of the extended SVBSDF method with the baseline photometricNet.	64
• 2.28 Microscale SVBSDF maps.	65
• 2.29 A full diagram of the model architecture of our extended photometricNet. . .	67
• 2.30 Ablation study of our proposed improvements with respect to photometricNet.	68
• 3.1 Results obtained with SeamlessGAN for tileable texture synthesis	69
• 3.2 Illustration of the capabilities of SeamlessGAN	72
• 3.3 Overview of SeamlessGAN	74
• 3.4 Training framework of SeamlessGAN	76
• 3.5 Impact of the latent field level on the quality of the outputs of SeamlessGAN.	78
• 3.6 Outputs of the discriminator of SeamlessGAN for textures of different qualities.	79
• 3.7 Examples of SeamlessGAN generating multiple tileable outputs from the same sample	80
• 3.8 Crops of synthesized textures using SeamlessGAN	82
• 3.9 Ablation study on the impact of the loss function on the quality of the synthesized textures	85
• 3.10 A comparison of the results of our proposed architectures	86
• 3.11 A comparison of our work with Texture Stationarization	88
• 3.12 Qualitative comparison of SeamlessGAN with different texture synthesis methods.	90
• 3.13 Examples of failure cases of SeamlessGAN	91

• 3.14 An ablation study on the influence of the number of layers in each model in the quality of the generated outputs.	95
• 3.15 More examples on the capabilities of SeamlessGAN on generating multiple tileable textures from a given input	96
• 3.16 Additional comparison with previous methods	97
• 4.1 Renders generated with NeuBTF materials	99
• 4.2 An overview of the inference and training processes of NeuBTF	102
• 4.3 Some tileable neural materials achieved with NeuBTF.	104
• 4.4 Qualitative results of NeuBTF on a variety of BTFs	106
• 4.5 A selection of latent channels learned by NeuBTF for a variety of materials..	108
• 4.6 Surface normals reconstructed from the materials encoded by NeuBTF	110
• 4.7 A failure case of NeuBTF	112
• 4.8 Examples of structural editions allowed by NeuBTF on a variety of materials	116
• 4.9 Multi resolution material generation using NeuBTF	117
• 4.10 diagram of the autoencoder of NeuBTF	118
• 4.11 A diagram of the renderer of NeuBTF	119
• 5.1 Teaser illustrating our low-cost optical capture pipeline	121
• 5.2 Illustration of the variance in rendering obtained by materials with different uncertainties	122
• 5.3 Scanner images vs fitted albedos	123
• 5.4 Overview of UMat	126
• 5.5 Top: input image of a rib material with metallic sequins. Bottom: α brdf and per-map uncertainties.	129
• 5.6 Qualitative results of some configurations of our ablation study	133
• 5.7 Embeddings of the transformer of our generator for data in the training set	133
• 5.8 Correlations between per-map errors and uncertainties	136
• 5.9 Illustration and results of our active learning experiment	136
• 5.10 Limitations cases of our method	139
• 5.11 A full diagram of the generator of UMat	144
• 5.12 A full diagram of the discriminator of UMat	145
• 5.13 Visualization of the dataset used to train UMat	146
• 5.14 Visualization of some Ground Truth SVBRDF of the different families in our test set.	147
• 5.15 Additional qualitative results of our uncertainty quantification method	148
• 5.16 Additional results of our uncertainty metric	149

• 5.17 Additional results of our active learning experiment	150
• 5.18 Results of our method on datasets of previous work	151
• 6.1 Teaser illustrating the capabilities of our capture setup and similarity metric for fabric mechanics.	155
• 6.2 Our proposed setup for capturing fabric mechanical parameter.	157
• 6.3 An overview of the main components of our learning-based mechanical parameter capture method.	160
• 6.4 Spearman correlation matrix between parameters of our synthetic dataset	162
• 6.5 Sweep of simulation parameters for hanging and stretch scenes	163
• 6.6 Diagram of our training and evaluation pipelines	165
• 6.7 Saliency maps of our proposed model, aggregated on target parameter	170
• 6.8 Human Judgments 2D tSTE embedding of human perceptual judgments about fabric similarity	172
• 6.9 Correlation between the ordering provided by the Human Judgments and our drape similarity metric	174
• 6.10 A comparison between the simulations obtained through the ground truth parameters and those obtained using the predictions of our model	176
• 6.11 Search by drape similarity	177
• 6.12 A failure case of our method	177
• 6.13 Relationship between the distance obtained by our user study, our distance metric and the difference in each parameter.	179
• 6.14 Illustration of our simulation-space data augmentation policy	179
• 6.15 Ranking classification accuracy on different ways of comparing materials	180
• 6.16 Correlation between the ordering provided by the Human Judgments and our Drape Similarity Metric.	181
• 6.17 Correlation between the ordering provided by our similarity metric with the Ground Truth parameters and the estimations	182
• 6.18 Correlation between the ordering provided by the tSTE embedding from our user study and our distance metric, computed using the estimated parameters.	183
• 6.19 Correlation between the ordering provided by the tSTE embedding from our user study and the distance computed on the parameter space, for the model predictions	184
• 6.20 Demographics of the participants of our perceptual study	184
• 6.21 Ablation study on different factors of our similarity metric	185
• 7.1 Renders using analytical illumination methods and our approach	187

• 7.2 Rendered images using traditional and our learning based illumination framework.	188
• 7.3 Overview of NEnv	191
• 7.4 Environment map and the PDF encoded by NEnv for an unprocessed and a rotated environment map.	192
• 7.5 Quantitative comparisons of the performance of different coupling layers for encoding environment map PDFs	200
• 7.6 A qualitative comparison between the type of coupling layer used in our normalizing flows	201
• 7.7 A visualization of the impact of the size of F	202
• 7.8 Environment maps encoded by our method and previous work.	203
• 7.9 Average seconds per sample for the baseline sampling algorithms and our model.	204
• 7.10 Render comparisons of different configurations of our work with the ground truth environment map and previous work	207
• 8.1 Renders with materials generated by some of the methods presented in this thesis	210

Tables

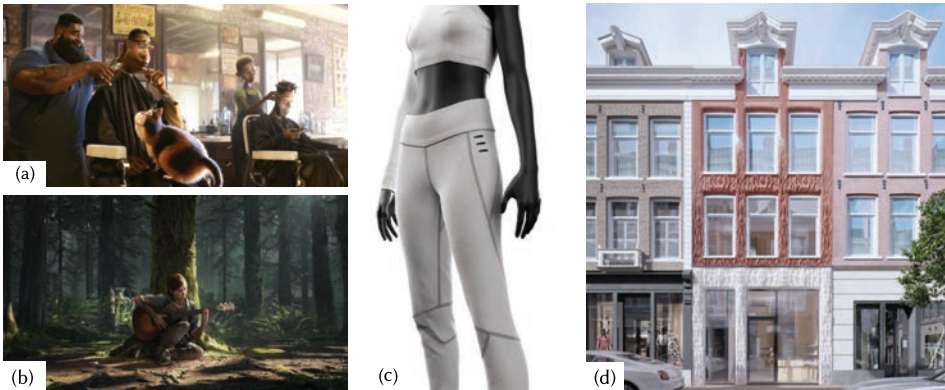
• 2.1 Quantitative comparison of photometricNet with previous work	48
• 2.2 Quantitative comparison between different number of blocks of layers in our model	52
• 3.1 Quantitative evaluation of the design of the generator	84
• 3.2 Quantitative comparison between different tileable synthesis methods	88
• 4.1 Number of trainable parameters in the decoders and amount of latent texture channels for different neural BTF compression algorithms	111
• 5.1 Results of our ablation study, across a variety of metrics	134
• 5.2 Model sizes for different methods, and their evaluation time in seconds	138
• 5.3 Comparisons of our method with previous work on SVBRDF capture	139
• 5.4 Comparisons of our method with previous work for a curdroy fabric	152
• 5.5 Comparisons of our method with previous work for a suede leather.	153
• 6.1 Ablation study of the neural architecture and data augmentation	166
• 6.2 Results for real depth images varying the input	169
• 6.3 Average correlation with rankings obtained through the tSTE Embedding	174
• 7.1 Average environment map reconstruction error for different methods	203
• 7.2 Average render reconstruction error for different methods	206

CHAPTER 1. INTRODUCTION

"We may say most aptly that the Analytical Engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves.

— **Ada Lovelace**

Figure 1.1: Renders of virtual scenes in different use cases. (a) A frame of Pixar's *Soul* (2020), directed by Peter Docter. (b) A frame of *The Last of Us Part II* (2020), Naughty Dog. (c) Garments designed with *SEDDI Author*. (d) A render of *Ceramic House*, courtesy of Studio RAP.

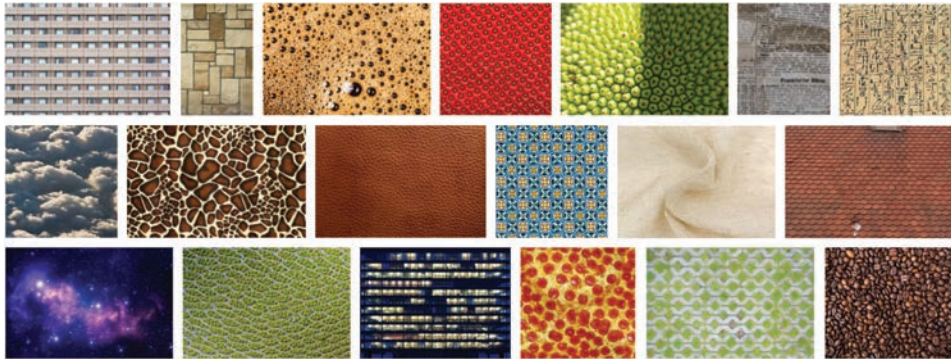


Generating lifelike virtual scenes has been a long-standing goal in computer graphics. This objective is now closer than ever before, thanks to the development of efficient methods for creating such scenes. Realistic virtual scenes have a range of applications, including media and entertainment, such as art, video games, augmented and virtual reality, and films. Furthermore, they are also important in industrial and commercial settings, such as computer-aided design, architecture, and fashion. We illustrate some use cases in Figure 1.1.

Achieving photo-realism in virtual scenes requires accurate and efficient representations of the optical behavior of materials, object geometry, and illumination. Although many methods have been developed for generating realistic imagery using these representations, producing photo-realistic images often requires a significant amount of manual labor from skilled artists, as well as substantial computational resources. One of the key bottlenecks in this process is the generation of digital representations of materials. Materials, depicted in Figure 1.2, are ubiquitous in the real world and exhibit an astonishing variety of structures, regularities, and colors. Realistically rendering them also requires modeling how they interact with light, taking into account their levels of glossiness, transparency, roughness, and microgeometric structures, all of

which impact their appearance. This wide range of material properties poses challenges to the development of efficient representations that can capture all the variations required to render the real world in a visually compelling way.

Figure 1.2: Examples of real-world materials and textures.



Furthermore, several design and manufacturing processes involve the manual design of materials, the outcome of which remains uncertain until different fabrication steps have been completed. This work methodology is ubiquitous in industries such as garment design or fashion, and it is slow and wastes a significant amount of valuable resources. The fashion industry alone is projected to generate more than 3 million metric tons of CO_2 emissions (10% of the annual worldwide emissions) and waste 170 billion cubic meters of water in 2025 [RSS16], significantly exacerbating the climate emergency. Advanced techniques in engineering and computing have opened up the possibility of digitally creating and editing simple materials, such as plastics or metals. However, complex deformable materials, notably fabrics, have not been extensively studied due to their intricate optical and mechanical properties. This is problematic, as billions of garments are manufactured every year (a staggering 80 billion in 2015 alone [Mor15], a number that has only increased since then). Besides, establishing a relationship between physical parameters that are relevant to the manufacturing process and perceptual parameters that closely align with how designers perceive materials is a challenging task. Achieving this mapping could be instrumental in enabling designers to modify materials according to their creative vision and effectively communicate their intentions to manufacturers.

Although plenty of solutions for material digitization exist, they are often limited in terms of accuracy, scalability, or cost. Accurately generating digital materials can be expensive and challenging, as it requires the use of inaccessible capture devices such as *gonioreflectometers*, or significant manual effort from skilled artists. These resource requirements not only lead to inefficiencies but also restrict the number of individuals who can benefit from material

digitization. While cost and time constraints are major obstacles to material digitization, data-driven algorithms and low-cost devices like smartphones offer promising solutions.

Offering digital solutions to the challenges of material digitization has the potential to bring significant benefits to various industries and society as a whole. These benefits include more efficient manufacturing processes, reduced waste, a more dynamic economy, and more immersive virtual experiences for art, media, and entertainment. In this thesis, we present innovative methods that address these challenges by leveraging ideas from computer vision, physically-based rendering, and recent advances in machine learning.

In the following sections, we provide a theoretical background of physically-based rendering (Section 1.1), the open problems in the literature (Section 1.2), our goals (Section 1.3) contributions to tackle these problems (Section 1.3.1) and a list of measurable outcomes of the contributions of this thesis (Section 1.4).

1.1 Theoretical Background

In this section, we introduce important theoretical background, which is relevant to the work executed in this thesis. *This background is derived from a survey on deep intrinsic image decomposition [Gar+22], which was written during the development of this thesis.*

1.1.1 Physically-Based Rendering

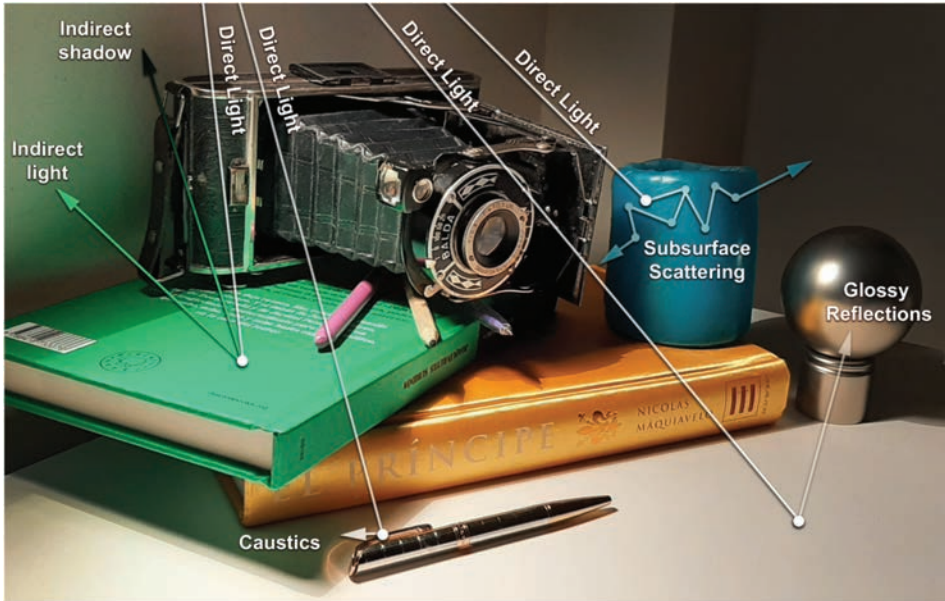
If we look at any simple scene surrounding us, such as the photography in Figure 1.3, we can find a plethora of optical interactions: indirect lighting (color bleeding), internal scattering in translucent objects, caustics, anisotropic and glossy reflections, etc. Far from the traditional assumptions in intrinsic imaging of diffuse (Lambertian) shading, direct lighting, and diffuse albedo materials.

In the following, we provide an overview of the theoretical background behind the image formation model, its derivation for non-diffuse materials, and the link with physically-based and inverse rendering. For a deeper dive into any of the concepts quite briefly described below, we recommend reading the book on physically based rendering by Pharr et al. [PJH20] and the survey by Guarniera et al. [Gua+19].

The color and luminosity at any point of an image, the incoming *irradiance*, is proportional to the sum of the outgoing *radiance* from all the visible points of the scene towards the camera sensor at the corresponding pixel, I , resulting

from multiple interactions between light and matter in the scene. Naturally, this is a simplification: even if we consider the camera lenses and color filters as part of the scene, the interaction of irradiance and the sensor point affects the result, and both electronic and film cameras have specific additional image formation steps which can be simulated. In physically-based rendering, the general approach to compute this value is to use Monte Carlo estimators of the pixel and shading integral [Kaj86], which has the general form of:

Figure 1.3: Example of light transport and material interactions. Secondary bounces of light produce reflection caustics (chrome pen) and color bleeding from the green book. The wax candle exhibits multiple internal (subsurface) scattering of photons. The yellow silk fabric of the book cover shows specular anisotropic reflections due to yarn orientation. Figure from [Gar+22].



$$I = \int_{\chi} f(x, \Theta) dx \quad (1.1)$$

where f is a function that defines the radiance towards pixel I and is defined on a domain χ , generally a unit sphere, or the set of all the surfaces (A) in the scene, and depends on the scene parameters Θ , which might include the definition of the geometry (normals, z-depth, vertices), material (albedo, BRDF), or illumination sources s far-field environment lighting, or 3D light emitters (point, area, objects).

The value of I is defined for a given λ , which is the spectral band of the camera sensor. We could define it for as many bands as desired (including non-visible

ones) but in practice, the majority of camera sensors mimic the human visual system and are commonly three-band: $\lambda \in \{R, G, B\}$. In subsequent rendering equations, we will simplify the notation by assuming a single-band, and omitting the term λ .

Equation 1.1 is an integral of integrals (see Figure 1.4): to account for all the light arriving at a surface point p_1 to a sensor pixel at p_0 , we have to estimate all the contributions of light from all the surfaces of the scene, recursively tracing paths bouncing in surfaces $(p_2, p_3 \dots p_n)$ until we reach the light emitted by a source $L_e(p_n \rightarrow p_{n-1})$. This is referred as the Light Transport Equation (LTE) in rendering and it is another way of seeing the equation 1.1. To compute the radiance reaching the pixel I , that is from p_1 to p_0 in Figure 1.4, we would need to solve:

$$L(p_1 \rightarrow p_0) = \sum_{k=1}^{\infty} P(\bar{p}_k) \quad (1.2)$$

with $P(\bar{p}_k)$ being the radiance scattered over a path $\bar{p}_k$ with $n + 1$ vertices $(p_0, p_1, p_2, p_3 \dots p_n)$ and computed as:

$$P(\bar{p}_k) = \underbrace{\int_A \int_A \dots \int_A}_{n-1} L_e(p_n \rightarrow p_{n-1}) T(\bar{p}_k) dA(p_2) \dots dA(p_n) \quad (1.3)$$

We can integrate over solid angles in the unit sphere, or surfaces (A) of the scene, being $dA(p_i)$ the differential area at point p_i . Note the term $T(\bar{p}_k)$, named *throughput* of the path: the fraction of radiance from the light source that arrives at the camera after all of the scattering at vertices between them. The total transmitted energy will be reduced at each interaction event, as some wavelengths (colors) are absorbed or scattered away from the observer. This is a common trade-off decision in many Monte Carlo rendering engines; longer paths per sample are costly to compute, while contributing less and less energy with each additional vertex, but in some scenes, they might be very relevant to reduce variance (noise) and converge with fewer samples to an accurate image.

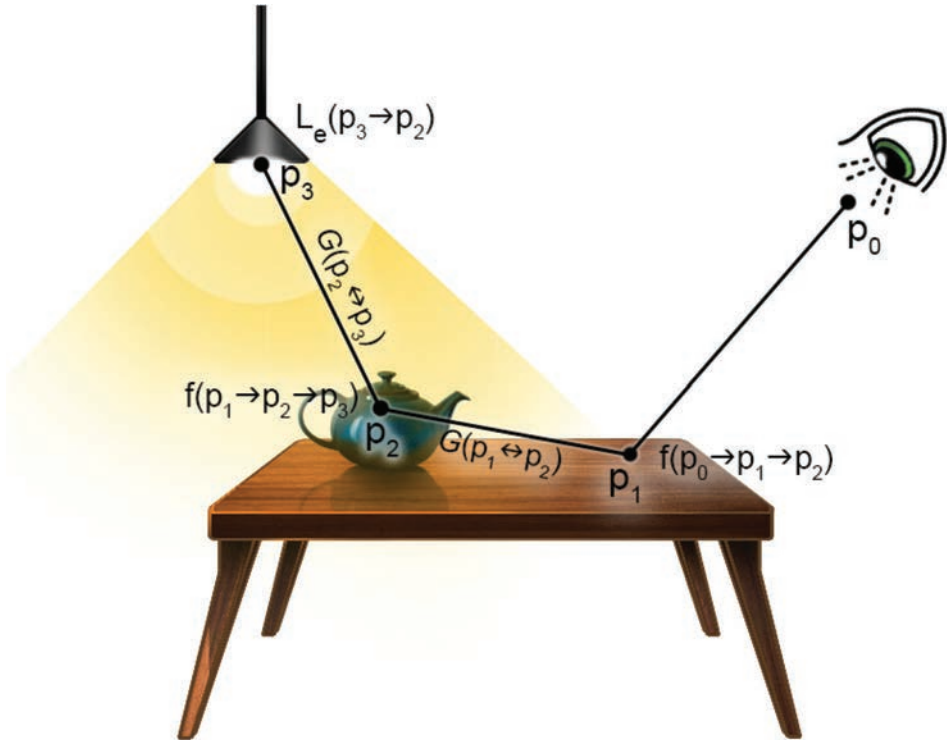
If there are interactions with non-opaque materials (human skin, cloth, marble) or participating media, such as liquids or smoke, we have to integrate volumetric scattering interactions of photons along the path between the light sources and the camera sensor pixel, requiring a more complex mathematical model such as the Radiative Transport Equation (RTE).

1.1.2 Geometry

If we take a look at Figure 1.4, we can observe that materials are distributed on discrete 3D objects: the table and the cup, and even the light source if

we consider it as an emissive material. We can assume that one of the most important Θ parameters is geometry, usually in the form of 3D vertices and edges, normals (N), or depth maps (D). Please note that, in contrast to a full 3D mesh, a single camera-view depth image (*a.k.a.* Z-buffer) is an incomplete definition because the non-visible surfaces are undefined, and the reflected light paths cannot be traced behind the visible objects. For example, Nimier et al. [Nim+19] require multiple views of a smoke volume in order to reconstruct its 3D density distribution by inverse rendering.

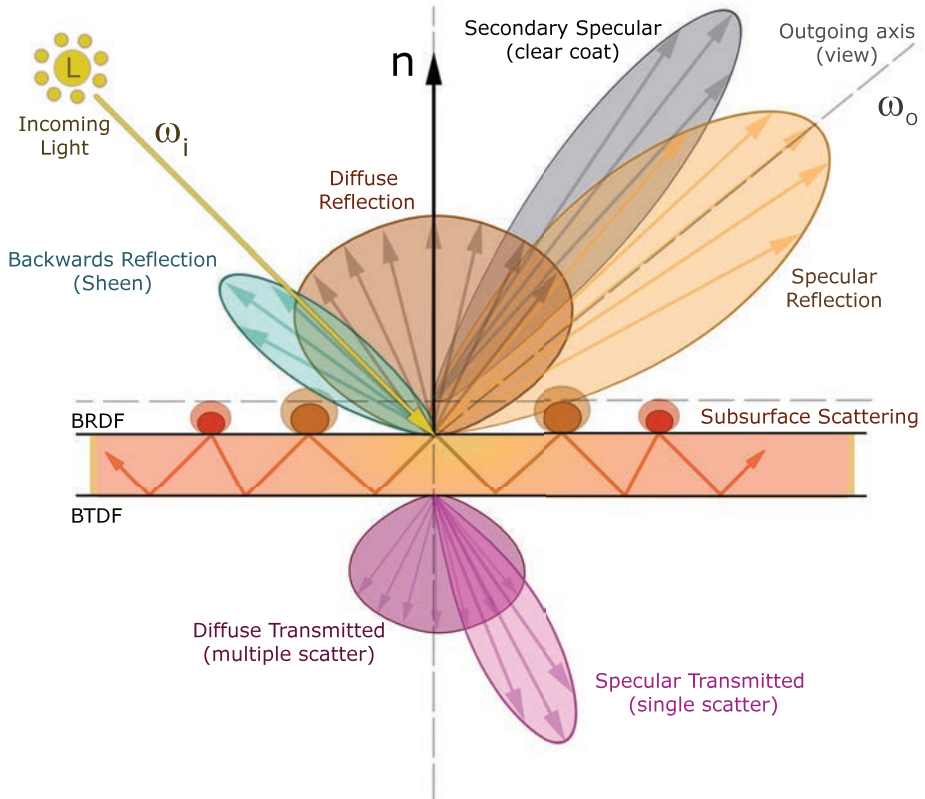
Figure 1.4: Example of light transport, connecting a light source p_3 to a pixel I at p_0 . Multiple paths like this one will need to be explored to provide a good statistical estimate of the radiance from p_1 to p_0 . Figure from [Gar+22].



There are additional ways of defining this geometry, such as implicit surfaces, but the material distribution is more complex when there are no clear surface boundaries. That is the case of heterogeneous participating materials: human skin, airlight, mixed liquids, smoke, etc., where light transport between two points has to be in turn evaluated along the path to account for all the possible scattering effects. For instance, imagine a small cloud of vapor between points p_1 and p_2 in Figure 1.4, at each infinitesimal step along the path between those points, there

will be a possibility of absorption (collision with a water particle) that will reduce the energy, but there is also a possibility of receiving incoming energy (not emitted by p_2), as the cloud itself is receiving direct illumination from the light source and multiple scattering events distribute the light across its volume.

Figure 1.5: 2D depiction of a physically-principled BSDF theoretical model. In most standard representations, the continuous reflectance 4D function is discretized into a combination of analytic lobes (Cosine, GGX) which can be easily computed sampled. Note that subsurface scattering (photons traveling through the medium) is not composed by multiple lobes. They are depicted for descriptive purposes, but it is rather approximated by a constant value or a diffusion profile, if not explicitly computed by simulating multiple scattering events with path tracing or photon mapping. Figure from [Gar+22].



In order to compute the rendering equation in volumetric media, a distribution of parameters is required at any point of the space, not only on the 3D surfaces. The usual representations include solid 3D implicit functions (for instance 3D Perlin noise, used in cloud procedural generation), meshes and distance fields,

or discrete volumetric grids which store the scattering probability function at any point of the space (also known as *phase function*) by means of voxels.

1.1.3 Materials

Each time the light interacts with a material, there is a loss of energy and a transformation of the original wavelength reflected towards the observed direction. In rendering, the result depends on the intrinsic material response for those two angles: incident light and viewing direction (e.g.: the camera, or another element of the scene). For *surfaces*, this response is modeled with a Bidirectional Reflectance Distribution Function (BRDF) $f_r(x, \omega_i, \omega_o)$ which yields at each particular 3D surface point x , and for each incident direction ω_i , the fraction of reflected radiance observed from a direction ω_o .

The total reflected radiance L at any point x can be obtained by integrating with Equation 1.4 over the positive hemisphere Ω^+ , to sample the whole incident light attenuated by the cosine term (dot product between the incident light direction L_i and the normal of the surface N) [Kaj86]:

$$L(x, \omega_o) = \int_{\Omega^+} f_r(x, \omega_i, \omega_o) L_i(x, \omega_i) (\omega_i \cdot N) d\omega_i \quad (1.4)$$

Note that by integrating the computed radiance $L(x, \omega_o)$ of the points sampled from p_o at the camera sensor ($x = p_i, \omega_o = p_1 - p_o$, see Figure 1.4), we are obtaining the irradiance at the sensor and the corresponding image pixel values (I in equation 1.1). Naturally, even the pixels themselves can be sampled several times and integrated over the camera sensor with another Monte Carlo estimator to minimize aliasing effects.

The BRDF can be extended with a Bidirectional Transmittance Distribution Function (BTDF) to conform a full Bidirectional Scattering Function (BSDF), defined in the full sphere Ω . The model can be further extended to account for Surface Scattering phenomena (BSSDF).

These functions have a minimum of four dimensions (input-output pair directions in polar coordinates) and usually three RGB values as output. It is thus technically possible to choose a discrete set of (ω_i, ω_o) orientations and create a lookup table to interpolate the response of the material, which is captured with multiple light and camera positions (e.g. with a gonioreflectometer). Storage becomes a major drawback for tabulated data, which can only be reduced through a significant reduction of quality. Moreover, we are considering only homogeneous surface materials, which is not often the case in actual scenes (e.g., a printed paper). Those spatially-varying values

(SVBRDF) can be stored in a stack of textures; or a Bidirectional Distribution Texture Function (BTF), increasing the dimensions and size of the table.

Beyond direct compression techniques, the most successful approach in graphics has been the use of analytic N-dimensional functions to approximate the reflectance and scattering distributions. The simplest of them, Lambertian diffuse, and Phong specular shading are also well known in the computer vision community. These functions leverage symmetry (isotropy) to model the material with a few parameters. For instance, a Lambertian material only requires knowing the intrinsic albedo, a sort of base color, while the Phong model requires three additional parameters for the specular component (e.g., shininess).

In the following subsections, we review the most common and simple material assumptions used in recent papers, finalizing with the most sophisticated inverse material models which are starting to be studied in computer graphics.

Lambertian Assumption

The Lambertian assumption is the most common material reflectance simplification used to tackle the problem of intrinsic image decomposition. It consists of assuming that the BRDF of a surface is constant in all directions (diffuse) and, consequently, the observed light radiance does not depend on the viewpoint. Therefore, we can omit ω_o in the surface reflectance model $\hat{r}$ used in Equation 1.4.

If the surface is diffuse, then $f_r(\omega_i) = \frac{\rho d}{2\pi}$, with ρd denoting the diffuse albedo: the constant ratio of incident light which is reflected in any direction, independently of the viewpoint ω_o . The image pixel value I is then given by:

$$I = \underbrace{\frac{\rho d}{\pi}}_A \underbrace{\int_{\Omega^+} L_i(\omega_i) (\omega_i \cdot N) d\omega_i}_S \quad (1.5)$$

The intrinsic model then can be defined as,

$$I = A \cdot S \quad (1.6)$$

where S contains all the shading variations due to the geometry of the local surface w.r.t. light direction. In some cases, the shading image should contain contributions of all the lights in the scene ($S_1 + S_2 + \dots + S_k$), which for discrete directional lights, can be deterministically estimated with a linear summation:

$$S = \sum_1^K L_i(\omega_i) (\omega_i \cdot N) \quad (1.7)$$

In the case of a more realistic illumination representation, such as environment lighting, or indirect light, the shading computation requires sampling the whole hemisphere Ω^+ , often recursively sampling other surfaces to approximate the integral of the incoming light. The shading component S within Equation 1.5 in the integral form, is difficult to compute and not so easily invertible and differentiable, so until recently, most intrinsic decomposition methods assumed the simpler formula described in Equations 1.6 and 1.7. Note that the illumination visibility is not considered in most cases (E.g.: cast shadows).

Non-Lambertian Assumption

There are two possible sources producing a Lambertian shading, either a surface which has an extremely rough micro-geometry, and thus reflects light equally in multiple random directions at any differential patch of the surface, or a very diffuse light source, coming from any direction with equal intensity (e.g., a foggy day). Both scenarios can be combined (Equation 1.4): the shiniest object on a foggy day will look quite diffuse, while even the most diffuse materials tend to project specular reflections under focused lighting from certain view angles. However, the majority of materials in the world are not Lambertian: even the most diffuse surface will exhibit Fresnel reflections when observed at grazing angles. Therefore, most surfaces will show the view-dependent effects that are classified as specular reflections. This separation between specular and Lambertian is rather pragmatic, but arbitrary, as even a simple microfacet model (shown in Figure 1.5) requires multiple analytic 3D lobes to approximate the 3D reflectance response for an infinitesimal incoming light ray (ω_i). The term *specular* is usually applied to narrow lobes with high probability of scattering radiance, producing high luminance values at pixels (highlights). This family of materials are of the general form:

$$L(\omega_o) = \int_{\Omega^+} \underbrace{f_r(\omega_i, \omega_o)}_{f_{NL}} (\omega_i \cdot N) L_i(\omega_i) d\omega_i \quad (1.8)$$

$$f_{NL}(\omega_i, \omega_o, N) = f_d(\omega_i, N) + f_s(\omega_i, \omega_o, N) \quad (1.9)$$

where f_{NL} is a non-lambertian BRDF composed by two components: f_d , a diffuse isotropic lobe, and f_s , a specular lobe which depends on the camera viewpoint (ω_o).

Dichromatic Reflection Model This particular Non-Lambertian model [MFS08; Tom94] separates the object in two reflection components (S_d, S_s), but considers that the specular component S_s might have a color α_s which could differ from the color of the reflected light:

$$L(\omega_o) = \alpha_d \int_{\Omega^+} \underbrace{f_d(\omega_i, N) L_i(\omega_i)}_{S_d} d\omega_i \quad (1.10)$$

$$+\alpha_s \underbrace{\int_{\Omega^+} f_s(\omega_i, \omega_o, N) L_i(\omega_i) d\omega_i}_{S_s} \quad (1.11)$$

$$I = \alpha_d S_d + \alpha_s S_s \quad (1.12)$$

This is the case of metallic materials which, unlike dielectric ones, will show specular reflections with a change in wavelength. Some additional effects such as colored interreflections might be also captured in all layers.

Phong and Blinn-Phong. The dichromatic model can be extended with one of the most adopted analytic approximations, either Phong (f_{NL}^P) or Blinn-Phong (f_{NL}^{BP}), which could be estimated with Monte Carlo integration and arbitrary lighting, or analytically computed with directional light sources:

$$f_{NL}^P(\omega_i, \omega_o, N) = \alpha_d(\omega_i \cdot N) + \alpha_s(h \cdot N)^k \quad (1.13)$$

$$f_{NL}^{BP}(\omega_i, \omega_o, N) = \alpha_s(\omega_i \cdot N) + \alpha_s(r \cdot v)^k \quad (1.14)$$

where α_d and α_s are the colors of the diffuse and the specular reflections, the halfway vector $h = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|}$ depends on the light direction ω_i and the view

direction ω_o . The size of the specular lobe is determined by the scalar term $k \in \mathbb{R}$.

Beyond Dichromatic Models: Physically-based Materials

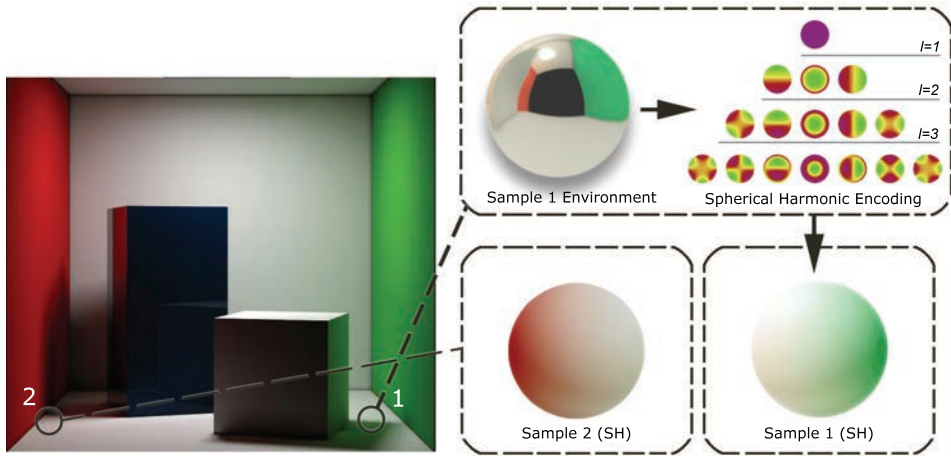
Naturally, the breadth of materials that can be synthesized with the previous models is very limited and not quite realistic in most cases. The advent of physically-based materials has introduced many variations [Hil+15] of the original microfacets models [TS67], which assume that a surface is composed of many very tiny facets that reflect light perfectly. By controlling the statistical distribution of their orientations, the roughness of the surface varies from mirror-like to almost diffuse. Additional optical properties are introduced in these models: Fresnel view-dependent reflectivity, multiple specular lobes, metalness (conductive materials such as gold, change the color of the highlights), multiple reflection and refraction lobes, etc.

The separation of the lobes described in the multi-lobed physically-based model shown in Figure 1.5 is not arbitrary, grouping reflections and refractions which share the same orientation and energy level. For instance, the main diffuse lobe is grouping multiple different orientations which are not view-dependent and share the same intensity and color. If it covers the full hemisphere with a cosine-like ratio, it is often referred as *Lambertian*. Likewise, the specular transmitted lobe is grouping a view-dependent peak that the observer would only see when the translucent surface is between the light source and the camera.

Even if it is often called a *single scatter* lobe, likely multiple internal scattering bounces of light are also included in this group.

Figure 1.6: Example of spatially varying illumination encoding with Spherical Harmonics (SH). The incoming lighting can be computed globally for the whole scene (far field), or locally, at multiple points (near field) as we show for the two samples near each colored wall. If we project the irradiance (top row) into an SH basis we obtain a diffuse low-frequency representation (examples in bottom row).

Figure from [Gar+22].



1.1.4 Illumination

The illumination is a significant contributor to the shading term (S) in most decompositions. From a rendering perspective, as shown in Equation 1.15, the computation of the pixel radiance, L , requires considering both the emitters L_e and the irradiance: the integral of all the incoming lighting at the observed point. This incoming illumination is often neglected, considering only point light or directional analytic emitters, which simplify the shading computation by removing L_i from the integral. If f_r is also assumed to be Lambertian, only the form factor given by the cosine of the surface normal, and the light direction remains.

$$L(\mathbf{x}, \omega_0) = L_e(\mathbf{x}, \omega_0) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_0) L_i(\mathbf{x}, \omega_i) \max(\omega_i \cdot \mathbf{N}, 0) d\omega_i \quad (1.15)$$

In actual scenes, the incoming lighting is a combination of emitted or reflected illumination from distant objects (far field) and local surfaces close to the observed area (near field). The former is usually approximated in computer graphics with environment lighting, often based in High-Dynamic-Range (HDR)

images mapped into an infinite sphere or cube surrounding the scene, while the latter can be derived from the far field illumination, by simulating the local secondary light bounces. If the geometry does not change, an environment map can be stored at multiple scene locations and distances, to include near field effects more accurately (Spatially Varying Environment Maps), although at a great memory cost, and only for static scenes.

To reduce the sampling and size of environment maps and simplify the computation, Ra-mamoorthi and Hanrahan [RH01a] proposed their compression with Spherical Harmonics (SH), a set of orthonormal basis functions defined on the spherical domain (elevation θ and azimuth ϕ angles). Thus the equation 1.16 describes the irradiance E as the sum of bases weighted by the cosine decay term A_i and the illumination coefficient $L_{i,m}$. By changing the representation of the bases $\hat{Y}$ to polynomial coordinates of a unit normal $\mathbf{n}=(x,y,z)^T$, this becomes an efficient vector dot product operation (Equation 1.17). With the required modifications, this strategy is feasible with other orthogonal basis functions on the sphere.

$$E(\theta, \phi) = \sum_{i,m} \hat{A}_i L_{i,m} Y_{i,m}(\theta, \phi) \quad (1.16)$$

$$E = \hat{Y}^T L \quad (1.17)$$

$$E = T^T L \quad (1.18)$$

If we want to account for near field occlusion and interreflection, it is possible to precompute those local interactions, because they depend on the object geometry and materials, and not on far field illumination. This family of techniques is known as *precomputed radiance transfer* (PRT) [SKS02]: they precompute multiple events of light transport (see Figure 1.4) into the T term in Equation 1.18 with Monte Carlo pathtracing. In this fashion, each pixel will have secondary light bounces stored in a light transport map. If only the visibility term $V(\omega_i)$ is considered for, the method will be storing the ambient occlusion shadows, but not colored interreflections.

In Figure 1.6, we can see a pyramid of spherical harmonics bases $Y_{i,m}$, with different coefficients. It is important to know that, although usually nine bases are considered enough to account for 99% of the far-field irradiance at diffuse surfaces, this percentage is significantly smaller in glossy surfaces (requiring many more coefficients). Moreover, a small number of coefficients will never account for high frequency effects, such as cast shadows from high-frequency light sources (E.g: a point light representing the sun), even producing ringing artifacts if we try to increase the accuracy by adding more bases. There are other popular basis in rendering such as Haar Wavelets or Spherical Gaussians (SG) [Wan+09], which also have very interesting properties.

1.2 Open Problems

In this section, we aim to address the challenges inherent in creating precise and efficient virtual representations of materials and scene radiance. We begin by discussing the current difficulties in acquiring and editing high-quality digital materials that are suitable for photorealistic physically based rendering. Specifically, we will explore the challenges associated with tileable texture synthesis and image-based material propagation.

We then turn our attention to the major roadblocks that hinder low-cost digitization of the mechanical and optical parameters of real-world materials. We will examine the various obstacles to overcome when attempting to accurately capture and reproduce the physical characteristics of materials in a virtual environment, using scalable and affordable pipelines.

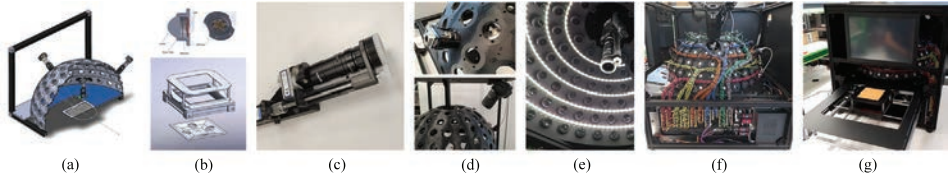
Lastly, we will review the limitations of current research on radiance encoding using neural networks, with a particular focus on representing material reflectance and scene illumination. By identifying these challenges, we hope to provide insight into the current state-of-the-art in material and radiance modeling for physically based rendering.

Digital Material Generation

Accurate and high resolution material reflectance digitization is a crucial problem for generating compelling and realistic virtual environments. As we have mentioned in Section 1.1.3, this digitization process is typically done using a *gonioreflectometer*, a highly complex machine designed to measure the photometric response of a material using light sources and cameras placed at different positions around the hemisphere. Existing devices for measuring material appearance in spatially-varying samples are limited to a single scale, either micro or mesoscopic. This is a practical limitation when the material has a complex multi-scale structure, like is common on many real-world materials. In many materials, accurately measuring their spatially-varying reflectance requires high resolution microscopic images. For example, the optical behavior of textiles is very dependent on small fibers or their yarn twist [CLA19]. Moreover, many materials also show large-scale *mesoscopic* variations, like prints, plaid, or tartan patterns, which cannot be captured using microscopic photographs. Further, many of such devices cannot capture important optical properties, like the material transmittance, which is key for the realistic rendering of objects like foliage or fabrics. There are no available capture devices or algorithms for accurate and high resolution SVBSDF digitization of heterogeneous materials. In order to introduce such materials, which are very common in many industries like fashion, textile, or leather manufacturing, into rendering pipelines, these machines need to be created and accurate algorithms need to be developed to handle the data

they capture. Finally, such a device may serve as a ground truth data generation source of training models that operate on more limited data scenarios, like in single image digitization

Figure 1.7: Schemes and photos of SEDDI's optical device, which can capture highly accurate and detailed SVBSDFs. (a) Schema of a cross-section of the hemisphere including micro camera and one polar camera. (b) (Top) Schema of the collimated LED design to account for the polarizer and collimating lens; (Bottom) Schema of the main holder and backlight support. (c) Microscopic optical setup. (d) (Top) A Polar Camera; (Bottom) Mid-distance Camera. (e) Interior of the dome with the diffuse LED strip activated. (f) Exterior of the dome and wiring. (g) Holder and exterior cover. Figure from [Gar+23].

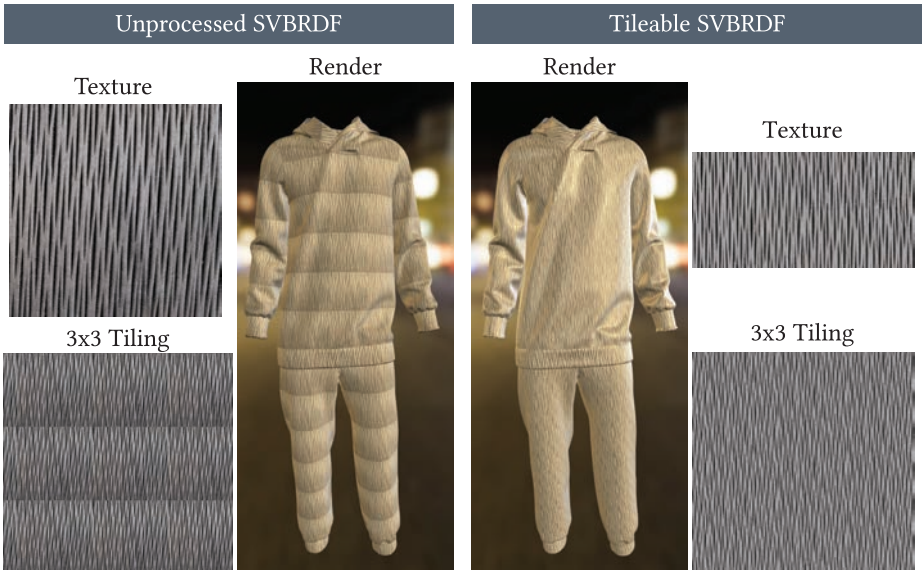


Tileable SVBRDFs Even if reflectance is accurately measured for a large portion of a material, in rendering settings, it is also important that the digital material is *tileable*. A tileable texture, also known as a seamless texture, is a texture image that can be repeated infinitely in all directions without any visible seams or discontinuities. In other words, a tileable texture is designed in a way that it seamlessly connects with identical copies of itself, which can be arranged side-by-side or stacked on top of one another to create a larger, continuous pattern without any visible repetition. This is often used in computer graphics, especially for creating textures on 3D models, or even for tiling backgrounds on websites or applications. We illustrate the importance of *tileable SVBRDFs* in Figure 1.8. Recent progress has been made in generating tileable textures from a single input example [Mor+17; Rod+19; Mor+20], however, these works present several shortcomings for tileable SVBRDF generation. First, they either assume a particular level of regularity, or the generated textures lose a significant amount of visual fidelity with respect to the input exemplars. Further, these methods are limited to generating a single texture map (i.e. the albedo of the material). This is problematic for tileable SVBRDF synthesis, which requires transforming every texture map in the SVBRDF into a tileable map, all while preserving pixel-wise coherence between maps. All in all, generating high-quality tileable SVBRDFs from a single input example is not currently possible with the methods proposed in previous work.

Material Attribute Transfer Effective material representations require understanding the properties that uniquely define them, which we refer to as *visual*

material attributes. These attributes are spatially-varying parameters that maintain spatial coherency with respect to the material structure, while remaining invariant to changes in the scene illumination or the geometry of the underlying object. For example, they may represent optical properties of a SVBRDF, artistic stylizations or higher-level properties, as in semantic segmentation masks. Obtaining these attributes for large samples of the material is very challenging. One way to address this problem is to obtain them at a small exemplar of the material, and *transferring* them into larger portions of it, using a large input image which serves as *guidance*. Different visual attribute transfer methods have been proposed in the past, however, they are limited in their efficiency, robustness, generalization or overall performance. Ideally, such a transfer method should be robust to new material illumination conditions, camera degradations or input distortions; as well as being efficient (providing interactive transfer times even in low computational budgets), controllable and predictable, and capable of generating high resolution outputs, which are required for realistic SVBRDFs. Developing a method with these characteristics may prove useful for many downstream tasks, like efficient data generation or material digitization pipelines.

Figure 1.8: On the left, an input, unprocessed SVBRDF, which is not tileable. On the right, a tileable version of the same material. Renders generated using SEDDI's [Textura.ai](#).



Scalable Digitization

Material Reflectance Estimation As mentioned, accurate material digitization typically requires expensive devices which take significant amounts of

time and manual input to operate. One such case is the capture machine illustrated in Figure 1.7, which is presented in [Gar+23]. These provide highly realistic digital representations of material reflectance at the cost of scalability, hindering their applicability in many real world settings. For instance, many manufacturing processes, like those in the textile industry, generate a massive variety of different materials, which cannot be easily nor cheaply digitized with the required cadence that characterizes these industries. Further, the limited availability of these devices introduces additional inefficiencies and economic and ecological costs, as physical samples of the materials must be shipped from the client to the institution holding the capture device.

Data-driven material reflectance estimation from low-cost devices is a long-standing problem in the literature [AWL13; AWL15; Des+18; Hen+21; Guo+20b; VPS21]. These methods promise to provide scalable, efficient, inexpensive on-site material digitization. They typically rely on one or more flash-lit images of a material taken with a smartphone camera, and a deep learning model trained on synthetic materials, which outputs plausible estimations. However, these approaches present several drawbacks that make them unsuitable for practical digitization workflows: First, many of such methods rely on generative models which tend to produce artifacts, or use large neural networks which puts limits on the output resolution, hindering the quality of the estimated materials. Besides, smartphone flash-lit images introduce significant calibration challenges; while most existing datasets used to train these models are purely synthetic, further limiting the quality of these estimations. Moreover, many of these methods use perceptual losses for training or in test-time optimization, which, while they help digitize stochastic materials, present additional challenges for guaranteeing the repeatability and consistency required for building a digital inventory. On top of these limitations, few-shot material reflectance estimation is still an ill-posed problem, as such, many different outputs may be plausible given the same input. So far, there is no way to quantify uncertainty for this problem, which hinders the applicability of these systems in real-world scenarios, and limits their trustworthiness, efficiency, and reliability.

Fabric Mechanical Behaviour Besides material reflectance, another important component of scenes is object geometry. While many real-world objects have static shapes, which can be scanned with relatively accessible devices like the LiDAR sensor in *iPhones*, there are materials that change shape when external forces are applied to them, like gravity. In this sense, *deformable materials* require particular computational models and capture pipelines. One of the most common of such materials is fabrics, which are ubiquitous in the real world, and used in our clothing, furniture, or vehicles. For them, accurately capturing their static shape is not enough: We need to be able to simulate their behavior on new scenarios, like new garments or dynamic forces, like wind [Ber+23] or human movement [SOC22]. Furthermore, textiles are incredibly varied in their fabrication patterns (eg different weave patterns or knitted structures), compositions,

and finishing, which not only determine their optical appearance [CLA19] but also their mechanical behavior, as shown in Figure 1.9. The problem of capturing fabric mechanical properties so they can be re-simulated in virtual settings has been amply studied in the literature, however, current solutions require tedious human intervention or expensive capture devices, hindering their scalability. Further, there is a lack of understanding of the perceptual similarity of fabric mechanical behavior, which creates additional challenges for designing accurate fabric mechanical parameter estimation pipelines.

Figure 1.9: Two real garments and their *digital twins*, for two fabrics with different mechanical behavior, as seen on their final drapes. Figure from [Rod+23b].

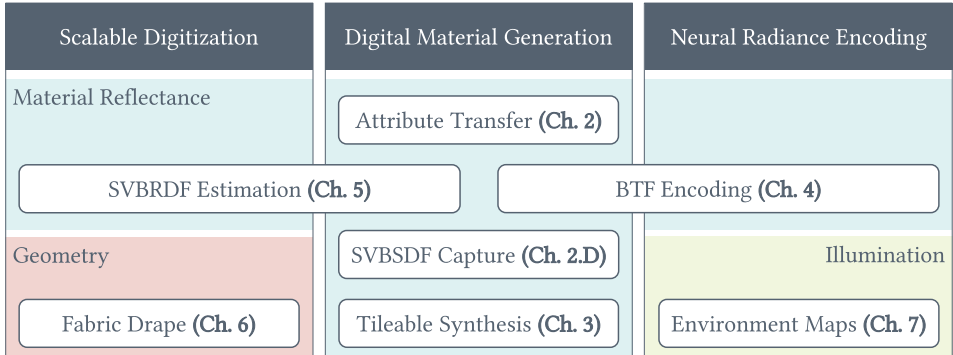


Neural Radiance Encoding

Reflectance In Section 1.1.3, we have mentioned several limitations of current representations of material reflectance. In particular, Bidirectional Texture Functions (BTFs) provide dense and accurate material reflectance measurements, however, they are typically prohibitive in terms of computational and memory cost. Recent work [Rai+19; Rai+20; Kuz+21] introduce different neural compression algorithms for BTFs, achieving remarkable realism and efficiency. Neural reflectance encoding methods trained on either synthetic or measured BTFs have thus shown promising results for increased realism of rendered materials. However, existing neural material encodings are immutable, meaning that their output for a certain query of UVs, camera, and light vector is fixed once they are trained. This can become very limiting when the fragment of the material used for training is too small or not tileable, which frequently happens when the material has been scanned with a capture device. Therefore, current neural reflectance encodings, while accurate and efficient, have

severe limitations in terms of editing capabilities, which hinder their applicability and usefulness in real world rendering settings.

Figure 1.10: Overview of the structure of this thesis. We propose different algorithms to solve problems (each column) in different components (each color) of virtual scenes.



Illumination Finally, as we mention in Section 1.1.4, there are many available approximations for representing real-world illumination in virtual scenes, including Spherical Harmonics, Spherical Gaussians, or Haar Wavelets. These, however, struggle to efficiently and accurately represent non-diffuse illumination conditions, like high-frequency light sources, as the sun or light bulbs. Recent work on neural representations of natural illumination [GES22] provides better approximations for such cases, but struggle with indoor lighting or night scenes with multiple light sources. Furthermore, these neural approximations do not provide sampling and PDF evaluation capabilities, which are essential for Monte Carlo rendering in path tracing. To the best of our knowledge, there are no lighting approximation methods that provide these sampling capabilities, while working accurately on any type of input environment map. Recent work on invertible generative models and implicit neural representations may provide a promising pathway to achieve these objectives.

1.3 Goals & Contributions

The primary objective of this thesis is to create innovative algorithms for computational representation and capture of materials and scene radiance. These are important for several reasons. These digital representations may be used in a wide variety of applications and problems, including computer graphics, video games, film production, virtual and augmented reality, or computer-aided design. They have the potential to significantly increase the visual realism and quality of the virtual environments created in these settings. Furthermore, the accuracy and efficiency of these methods directly impact the quality and computational resources required for these tasks. As such, obtaining accurate and efficient

representations and digitization algorithms for materials and other scene components can help reduce the computational cost of rendering, design, capture, and modeling, which can lead to faster, more efficient, and scalable pipelines. Moreover, they may enable efficient procedural dataset creation for training models to solve downstream tasks, like dense segmentation, image or video generation, or embodied vision. Finally, besides efficiency and accuracy, additional features like differentiability or uncertainty quantification capabilities create opportunities for inverse rendering applications or active learning.

To this end, we introduce new learning-based methods to address the open problems described before. Our proposed solutions strive to achieve maximum computational and data efficiency, as well as accuracy, control and reliability. Furthermore, our algorithms are designed to be beneficial to end-users by incorporating human requirements such as perceptual components, edition, predictability, robustness, and uncertainty quantification. All of our proposed methods leverage neural networks, incorporating recent advances in implicit representations, architecture and loss function design, and training procedures. They are all fully differentiable, and may be incorporated into optimization pipelines to solve inverse rendering problems. Our solutions for radiance encoding may be seamlessly incorporated into path-tracer engines for increased render efficiency, or into neural scene representations for increased realism. Further, our high quality casual digitization methods allow for scalable and low cost material capture, which can help end users create their own realistic virtual materials, or create large datasets at a lower cost. We provide an overview of this thesis in Figure 1.10.

Our goals in this thesis tackle the aforementioned challenges in digital material generation, scalable digitization, and radiance encoding, and can be summarized as follows:

- Design new learning-based methods for predictable, controllable and high quality digital material propagation, generation, edition, and synthesis.
- Introduce novel efficient and differentiable representations for scene radiance.
- Democratize high-quality material capture by presenting new methods which are low cost, scalable, accurate, reliable, and perceptually-validated.

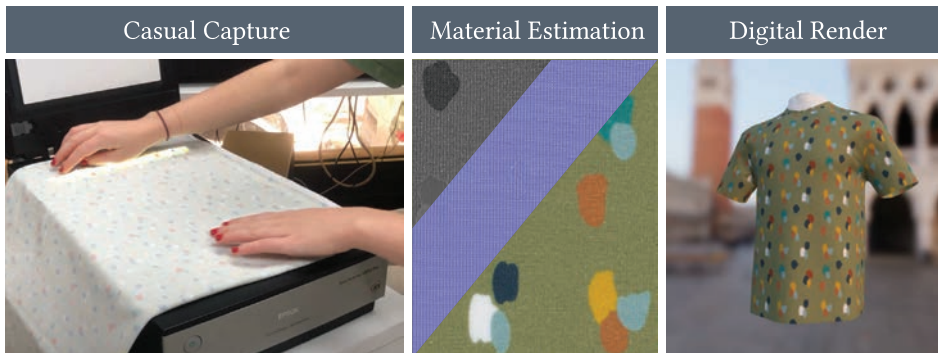
1.3.1 Contributions

The work developed in this thesis has led to the following contributions to achieve the goals presented above:

- A deep-learning based method for propagating spatially-varying attributes of a material to larger samples of the same or similar materials. At the core of this method is a lightweight fully-convolutional neural network trained using a *photometric dataset* and an extensive data augmentation policy. These contributions allow our models to generalize to new illumination, color, and geometric conditions. We show the effectiveness of our system on transferring attributes of different types, including surface normals, semantic segmentation, and artistic editions (Chapter 2).
- An extension to the previous method, for enabling a dual-scale capture system capable of digitizing a single material at high-resolution and accuracy levels. To do so, we extend the method presented in Chapter 2 for allowing to transfer multiple property maps at the same time, and with a higher degree of accuracy (Chapter 2.C).
- A generative model capable of creating *tileable textures* using a single image as input. By leveraging state-of-the-art generative adversarial networks (GANs) for texture synthesis, a novel latent space manipulation algorithm, and using the discriminator as a quality metric, our proposed model can generate tileable textures of higher perceptual quality, and at a lower cost than previous work. Further, we propose an extension for generating tileable SVBRDFs from a single input, and show the effectiveness of our method across textures of different levels of regularity. (Chapter 3).
- A neural field representation capable of *BTF encoding and transfer*. Building upon previous work on neural material representations, we propose a lightweight latent representation that can be decoded by an implicit neural network into per-texel reflectance values. This latent representation is estimated by an autoencoder-like network, which can be used to propagate BTF values to novel structures, allowing for neural material edition and transfer, and providing new capabilities for these types of representations. (Chapter 4).
- A novel single-image capture system capable of generating high-resolution SVBRDFs of materials captured from commodity scanners. Leveraging a custom-built dataset and a novel attention-enhanced image-to-image translation generative model trained with a variety of loss functions, our model provides artifact-free, highly accurate digitizations, using solely microgeometry patterns as cues. We further propose the first uncertainty quantification algorithm for SVBRDF estimation methods, building upon Bayesian deep learning approximations using Monte Carlo dropout during test time, and perceptual BRDF metrics. We show the effectiveness of our uncertainty metric for predicting digitization error at test time and for an active learning experiment which shows that uncertainty sampling helps build more data-efficient capture systems. (Chapter 5). We illustrate the capabilities introduced by this model in Figure 1.11.

- A learning-based system for digitizing mechanical properties of fabrics using a casual capture setup and a commodity depth camera. Training solely on synthetic data and leveraging an attention-enhanced multi-image neural network, transfer learning, and an extensive data augmentation policy, our proposed method can accurately estimate mechanical properties of fabric samples, without the need for expensive capture equipment. Further, we propose a novel image-based perceptual metric to measure mechanical similarity between fabrics, which we validate with a user study. (Chapter 6).
- A novel neural method for environment maps, which provides an efficient global illumination representation. At the core of our method lie two neural networks: a *normalizing flow*, capable of learning the *pdf* of an input environment map, and to efficiently sample directions from it; and a *implicit neural network* with sinusoidal activations capable of mapping from directions to linear RGB radiance values. We show the effectiveness of our method on *Multiple Importance Sampling* applications in rendering, obtaining accurate illumination at a significantly lower cost than traditional methods; as well as better quality environment maps than previous work on illumination representations. (Chapter 7).
- A comprehensive study of the capabilities and limitations of deep neural networks for *intrinsic image decomposition*. With a careful review of previous work, approaches, assumptions, datasets, losses, and neural network architectures, we provide a new categorization of these works. We further propose future research directions, building upon recent work on differentiable and inverse rendering and advances in machine learning systems. This contribution has been published in [Gar+22], a part of it is used in Section 1.1.

Figure 1.11: An illustration of the material digitization process enabled by some methods presented in this thesis. We show a picture of a user casually scanning a piece of fabric (left), the estimated SVBRDF for that material (middle), and a digital render of this SVBRDF (right). Assets generated using SEDDI Textura.ai.



Each of these contributions has been developed in collaboration with other researchers and engineers, many of whom are listed as authors of the publications listed below. However, most of this thesis would not have been possible without the outstanding work of the whole team at SEDDI, which has played roles in important tasks, including administrative work, dataset creation, financial support, help with software and infrastructure, as well as valuable scientific discussions. My individual contributions to each project can be inferred from my relative position in the author list.

1.4 Measurable Outcomes

1.4.1 Publications & Patents

Peer Reviewed Publications

The contributions in this thesis have led to the following publications, listed in chronological order:

- *“Neural Photometry-guided Visual Attribute Transfer”*
Carlos Rodriguez-Pardo, Elena Garces; *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2021 [RG21].
This journal has an impact factor of 5.226, and its position in the JCR index is 13 out of 110 (Q1) in the category COMPUTER SCIENCE, SOFTWARE ENGINEERING (data from 2021).
- *“A Survey on Intrinsic Images: Delving Deep into Lambert and Beyond”*
Elena Garces, Carlos Rodriguez-Pardo, Dan Casas, Jorge Lopez-Moreno; *International Journal in Computer Vision (IJCV)*, 2022 [Gar+22].
This journal has an impact factor of 13.369, and its position in the JCR index is 10 out of 145 (Q1) in the category COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (data from 2021).
- *“SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps”*
Carlos Rodriguez-Pardo, Elena Garces; *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2022 [RG22].
This journal has an impact factor of 5.226, and its position in the JCR index is 13 out of 110 (Q1) in the category COMPUTER SCIENCE, SOFTWARE ENGINEERING (data from 2021).
- *“How Will It Drape Like? Capturing Fabric Mechanics From Depth Images”*
Carlos Rodriguez-Pardo, Melania Prieto-Martin, Dan Casas, Elena Garces; *Computer Graphics Forum (Proceedings of Eurographics 2023)* [Rod+23b].
This journal has an impact factor of 2.363, and its position in the JCR index is 53 out of 110 (Q2) in the category COMPUTER SCIENCE, SOFTWARE ENGINEERING (data from 2021).

- *"UMat: Uncertainty-Aware Single Image High Resolution Material Capture"*
Carlos Rodriguez-Pardo, Henar Dominguez, David Pascual, Elena Garces;
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023 [Rod+23a].
This conference is the highest impact venue in computer science according to *Google Scholar*, and is rated as a CORE A++ conference (data from 2021).
- *"Towards Material Digitization with a Dual-scale Optical System"*
Elena Garces, Victor Arellano, Carlos Rodriguez-Pardo, David Pascual, Sergio Suja, Jorge Lopez-Moreno; *ACM Transactions on Graphics (Proceedings of SIGGRAPH), 2023* [Gar+23]. This journal has an impact factor of 7.403, and its position in the JCR index is 9 out of 110 (Q1) in the category COMPUTER SCIENCE, SOFTWARE ENGINEERING (data from 2021).
- *"NeuBTF: Neural Fields for BTF Encoding and Transfer"*
Carlos Rodriguez-Pardo, Konstantinos Kazatzis, Jorge Lopez-Moreno, Elena Garces; 2023
This paper is currently under peer review.
- *"NEnv: Neural Environment Maps for Global Illumination"*
Carlos Rodriguez-Pardo, Javier Fabre, Elena Garces, Jorge Lopez-Moreno; 2023
This paper is currently under peer review.

Patents

Some of the work developed during my PhD has resulted in two patent applications:

- *"Generation of macro-scale material property maps from images and micro-scale proper-ties"*
Carlos Rodriguez-Pardo, Elena Garces; (*SEDDI 2020, Filed Patent Status*)
- *"Neural synthesis of tileable textures"*
Carlos Rodriguez-Pardo, Elena Garces; (*SEDDI 2020, Filed Patent Status*)

1.4.2 Awards & Other Merits

We include here a list of awards and other merits received throughout the development of this thesis.

- *Top Reviewer* recognition at the 2022 Conference on Neural Information Processing Systems (NeurIPS).
- *Outstanding Reviewer* recognition at the 2022 European Conference on Computer Vision (ECCV).
- Our work on *Neural Visual Attribute Transfer* [RG21] (Chapter 2) was invited to the *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*.

- Our work on Generative Models for Tileable Texture Synthesis [RG22] (Chapter 3) was presented at *Machine Learning Tokyo* as an invited talk, and as a poster presentation at the AI for Content Creation Workshop (AI4CC) at CVPR 2022.

1.4.3 Student Supervision

During the development of this thesis, I have co-supervised the bachelor's final thesis of four computer science students:

- *"Multimodal Exploration of Textile Databases Based on Perceptual Parameters"*
María Pilar Alcarria Peinado, 2021. *Final grade: 94/100*
- *"A Fashion Recommender System Based on Multimodal Attributes"*
Gonzalo Llosa, 2023. *(Ongoing)*
- *"Image Generative Models for Garment Design"*
Alfonso Chiclana, 2023. *(Ongoing)*
- *"Web Interfaces for the Exploration of Text-to-Image Generative Models"*
Carlos Cepeda, 2023. *(Ongoing)*

1.4.4 Teaching, Talks, and Seminars

During the first half of 2023, I worked as an adjunct lecturer at Universidad Carlos III in Madrid, where I taught and led a mandatory graduate course on *Intelligent Data Analysis*, at the MSc in Computer Engineering.

Throughout my Ph.D., I had the opportunity to disseminate my work across different formal and informal events. I presented some of the work in this Ph.D. thesis on different venues, including:

- *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, 2022: I had the chance to present the work in Chapter 2 on propagating spatially-varying attributes of materials, as an invited talk of our TVCG paper.
- *Machine Learning Tokyo init*, May 2022: I presented the work in Chapter 3 on tileable texture synthesis as an invited seminar.
- *AI4CC workshop at CVPR*, June 2022; & *ELLIS Doctoral Consortium*, September 2022: I presented the work in Chapter 3 on tileable texture synthesis as poster presentations.
- *EUROGRAPHICS*, May 2023: We will present the work in Chapter 6 on casual and perceptually-validate mechanical parameter capture of fabrics.

- *CVPR, June 2023*: We will present the work in Chapter 5 on uncertainty-aware high resolution SVBRDF digitization.

1.4.5 Reviewing Activity

During the development of this thesis, I have served as a reviewer at multiple conferences, workshops, and journals on computer vision and machine learning:

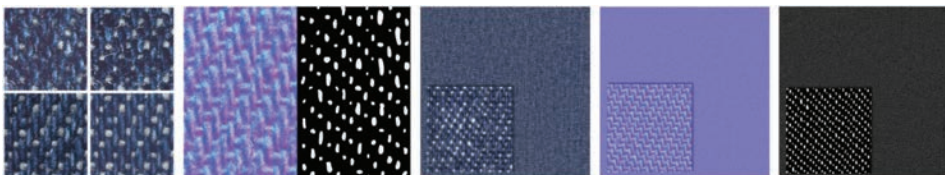
- *British Machine Vision Conference (BMVC)*: 2020
- *European Conference on Computer Vision (ECCV)*: 2022, Outstanding Reviewer
- *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: 2020, 2022, 2023
- *IEEE International Conference on Computer Vision (ICCV)*: 2021, 2023
- *International Conference on Learning Representations (ICLR)*: 2022
- *International Conference on Machine Learning (ICML)*: 2023
- *LatinX in Computer Vision Research Workshop*: 2023
- *Neural Information Processing Systems (NeurIPS)*: 2022 Top Reviewer, 2023
- *The Visual Computer Journal (TVCJ)*: 2020-2022
- *Women in Computer Vision Workshop (WiCV)* : 2022, 2023

1.4.6 Industrial Research Projects and Commercial Products

This thesis has been developed within an industrial Ph.D. in SEDDI, a startup based in Madrid, Spain, which has the goal of providing deep tech solutions to the textile and fashion industries. A significant part of the work presented in this thesis is part of commercially-available products and industry research projects. Notably, some of the algorithms (or variations thereof) presented in Chapters 2, 2.C and 5 are part of [Textura.ai](#), an AI-powered commercially available material digitization product. Further, the *drape similarity metric* presented in Chapter 6 has also been a part of a project developed for a textile manufacturer.

CHAPTER 2. VISUAL ATTRIBUTE TRANSFER WITH PHOTOMETRY-GUIDED NEURAL NETWORKS

Figure 2.1: Illustration of the capabilities of our attribute transfer framework. On the left, we show a subset of our *photometric dataset*, comprised of several images of a material captured with different illumination conditions. On their right, two visual attributes: surface normals and segmentation maps. Using a larger image of the same or similar materials as *guidance*, our method can propagate the visual attributes, achieving high resolution, robust and controllable transfers at small computational cost.



This chapter presents a deep learning-based method capable of propagating spatially-varying visual attributes of a material (e.g. texture maps or stylizations) to larger samples of the same or similar materials. By leveraging the material's photometric response and a comprehensive data augmentation policy, we make the transfer robust to novel illumination conditions and affine deformations of the underlying surface. Our model relies on a supervised image-to-image translation framework and is agnostic to the transferred visual domain; we showcase a semantic segmentation, a normal map, and a stylization. Following an image analogies approach, the method only requires the training data to contain the same visual structures as the input guidance. Our approach works at interactive rates, making it suitable for material edit applications. We thoroughly evaluate our learning methodology in a controlled setup providing quantitative measures of performance. Last, we demonstrate that training the model on a single material is enough to generalize to materials of the same type without the need for massive datasets. The contributions presented in this chapter have led to the following publication¹, which was also presented at the 2022 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D 2022) as an invited talk:

¹Additional publication details and results are included on [the project website](#).



"Neural Photometry-guided Visual Attribute Transfer"

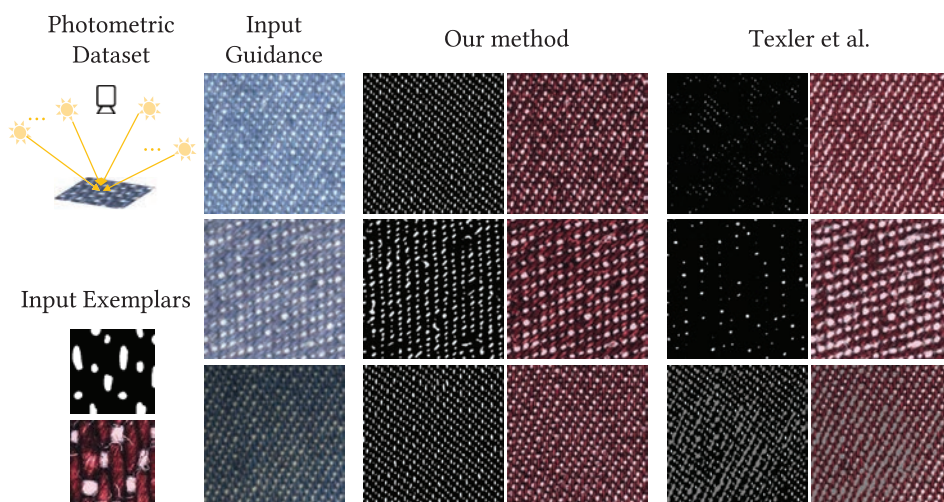
Carlos Rodriguez-Pardo, Elena Garces

IEEE Transactions on Visualization and Computer Graphics (TVCG)

(2021)

2.1 Introduction

Figure 2.2: The input exemplars on the left-bottom are transferred to the input guidances (second column) using images of the material taken under multiple illuminations (photometric input) as training data. Using this data for training, Texler *et al.* [Tex+20b] fail to generalize to geometric distortions and illumination conditions not present in the training set.



The development of effective and editable material models is becoming increasingly important so that users of virtual prototyping, video-games or AR/VR applications can have compelling and realistic experiences. For instance, allowing for the creation of virtual environments that surpass the uncanny valley, or empowering artists to create breathtaking visual settings. Effective material representations require understanding the properties that uniquely define them, which we refer to as *visual material attributes*. These attributes are spatially-varying parameters that maintain spatial coherency with respect to the material structure while remaining invariant to changes in the scene illumination or the geometry of the

underlying object. For example, they may represent optical properties of a microfacet spatially-varying BRDF (albedo, normals, roughness, anisotropy, etc.), but also artistic stylizations or higher-level properties, as in semantic segmentation masks. The former can be obtained with specific optical capture devices and estimation methods, either using machine learning [Mer+19] or classic optimization methods [Ste+14]. The latter may require time-consuming human labor.

Obtaining these attributes for large material samples can be problematic in certain situations. For instance, measuring high-resolution surface normals of large material samples or generating densely annotated segmentation masks, require either unfeasibly large capturing setups or expensive manual input. A common strategy to handle these cases consists of propagating these attributes using a larger image of the same material used as guidance image. This problem has been formulated within the context of image analogies [Her+01]. Propagating, or transferring, these properties requires preserving the local spatial regularities of the material, as well as adapting to the global variations of the guidance image, which is a highly challenging problem.

Previous work has used PatchMatch-based synthesis [Mel+12], look-up-tables [RPG16], or neural networks [Maz+19] by looking for repetitive patterns, match color statistics, or image gradients. These methods are however prohibitive due to their runtime performance, taking hours. Image analogies methods have been extended so as to leverage the power of convolutional neural networks as image descriptors [Lia+17; Ben+21], but those methods do not yield predictable outputs, and their computational complexity yields them impractical in interactive applications. Recently, neural networks have proven successful for the task of video stylization [Tex+20b], where the user needs to provide a few input editing exemplars which are propagated at interactive rates to the rest of the video. We show that this approach lacks the generalization power to work with images taken under unseen resolutions and geometric or illumination conditions.

In this chapter, we propose a novel learning-based method to propagate any visual material attribute estimated locally for a material's to larger samples of it. We train a neural network per material using image-to-image translation methods, making use of a policy of data augmentation that makes the transfer invariant to affine transformations (scale, rotations and shears). As opposed to other methods which blindly apply data augmentation strategies, our method is robustly evaluated using a real dataset to guarantee predictability in the estimations. Further, illumination invariance is obtained by feeding the network with multiple images of the material taken under a diverse set of illuminations. Our models can be trained in less than a minute and generalize to materials with the same microstructure.

In summary, we present the following contributions:

- The first method to use photometric data to train an image-to-image translation model capable of propagating any kind of visual material attribute to larger samples of the material, regardless of the illumination conditions of the input image.
- A data augmentation policy, thoroughly evaluated with a real dataset, designed to make the transfer invariant to affine deformations.
- Exhaustive comparisons with related work, demonstrating that we can achieve more predictable and higher-quality mappings with a fraction of the computational cost.
- We show that our trained models generalize to materials with similar microstructure as the ones used for training.
- On Section 2.C, we show an extension on this method for High Resolution SVBRDF capture, which is part of a publication accepted to *SIGGRAPH 2023*.

2.2 Related Work

Several computer graphics and vision problems are closely related to our method. The most similar ones are those related to any form of by-example *visual attribute transfer* (e.g. color, texture, style, or geometry). Besides, we also review material estimation and capture methods.

2.2.1 Visual Attribute Transfer

Visual attribute transfer refers to the problem of transferring some visual attributes (e.g. color, style, texture, or geometry) of one or many exemplars to another exemplar while preserving its *content*.

This problem can be formulated within the context of *Image Analogies* [Her+01], in which the goal is to stylize a target un-stylized image B , given a pair of images A (un-stylized) and A' (stylized). The most common approach to tackle this problem has been via patch-based texture synthesis [Bén+13; Bar+15; Jam+19]. More recent approaches have leveraged the capabilities of deep latent spaces within convolutional neural networks to disentangle style from content [Ree+15]. A seminal work by Gatys *et al.* [GEB15b] utilizes a VGG-19 convolutional neural network [SZ15] pre-trained on ImageNet [Den+09] as a feature descriptor for images, in which style and content are associated with different layers of the network, and transferred by gradient descent optimization. Their work on *style transfer* has been extended for single images [HB17; Li+17b; JAF16; Che+17b] and video [Che+17a], as well as for developing image-space distance metrics which resemble human perception [Zha+18]. A shared limitation of many of these

methods is their narrow capabilities to provide predictable edits, and considerable focus nowadays is put towards this end [Gat+17; Gu+18]. A comprehensive review on the topic of neural style transfer is provided by Jing et al. [Jin+19]. Our work differs from traditional style transfer approaches in the sense that we deal with a more constrained problem that requires predictable outcomes.

Exploiting the power of deep neural networks in the image analogies problem was tackled by Liao *et al.* [Lia+17] who, by assuming a semantic prior over an exemplar input image and a target one, propose a method capable of finding a bijective mapping between both inputs, enabling two-way stylizations. Single-image generative models [SDM19] were extended to the image analogies problem in [Ben+21], by using convolutional neural networks to generate a new image with the style of an input *style* and the *structure* of another image. These methods, however, rely heavily on content or semantic features, making them vulnerable to lighting or geometric differences between the input images; and are computationally expensive, rendering them impractical for interactive applications. Our method is robust to both geometric distortions and illumination variations, and works at interactive rates. Similar in spirit to our method, as it explicitly considers texture variations due to illumination, is the work of Fišer *et al.* [Fiš+16], which applies patch-match to provide illumination-dependent exemplar-based stylizations to cartoon pictures. In contrast, our work is meant to be illumination-invariant. Also concerned with stylization problems, Texler *et al.* [Tex+20b] present a method highly related to ours. They apply patch-based training of an encoder-decoder deep neural network using key frames stylized by a user. Resembling our approach, their algorithm also follows a few-shot learning strategy using as training data a few exemplar patches. However, as opposed to our method, they do not account for the variability of the appearance of materials under different lighting and viewpoint so, as we show in Section 2.7, their method does not generalize to unseen illuminations or geometric variations.

Image colorization is concerned with colorizing a gray-scale image given a few colorized exemplars. In this problem, it is critical to infer semantic relationships between the images so that the new scene is perceptually coherent and plausible [He+18; Zha+19a; He+19]. Similarly, *edit propagation* methods [AP08; End+16] work by propagating strokes provided by a user to the rest of the image, removing the need for a semantic understanding of the input image. Our work is related to the latter techniques, as we leverage the feature spaces of the CNNs and do not require a large labeled dataset to effectively solve our problem, and it can also be used to propagate segmentation masks [LAA08].

2.2.2 Textured Materials

Many real-world materials show spatial regularities, commonly referred to as *textures*. The patterns present in textures can be parameterized, which

allows for low-cost material capture or synthesis models. A way to model textured materials is through BTFs (Bidirectional Texture Functions) [Dan+99; LM01], a technique that uses multiple camera views and lighting angles to capture a dense sampling of the appearance of a material. Inspired by such methods, we leverage several images of the material under different illumination conditions, however, we require fewer data than typical BTFs capture setups [Rai+19; Rai+20]. Similarly, the problem of extrapolating BTFs captures to larger material samples was addressed by Steinhausen et al. [Ste+15a; Ste+15b], who propagate measured BTFs using texture synthesis. Our method is not meant to propagate full BTFs measurements but could potentially be applied to such datasets.

The goal of texture synthesis is to reconstruct a larger image given a small sample leveraging structural content. This is a long-standing problem in the computer graphics field and different strategies have been proposed, for instance, using PatchMatch [Dia+15], texture transport [AWL15], point processes [Gue+20; LH06], or neural networks [EM17; Zho+18; FAW19; Rod+19]. Also related to our work, Li *et al.* [LPG19] capture the appearance of materials by first estimating their BRDF and, then, synthesizing the high resolution microstructure from a dataset of measured SVBRDFs. Our problem is unlike texture synthesis, as we do not aim to create novel content but to predictably transfer visual material attributes.

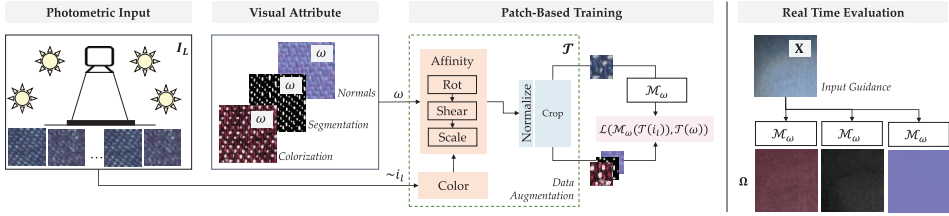
2.2.3 SVBRDF Estimation

The problem of estimating a SVBRDF model from one or several images using lightweight capture setups is becoming increasingly popular in the literature. Early work [HS05; HS03] leveraged Photometric Stereo [Ike81] and SVBRDF manifold bootstrapping [Don+10] for surface geometry reconstruction, while newer methods exploit the power of deep neural networks. Recent surveys by Guarnera [Gua+16] and Dong [Don19] contain relevant approaches. While our method is not meant to estimate the SVBRDF properties of a material, it can be used in combination with those techniques to create larger material assets.

There are a few methods that follow a similar paradigm to ours, transferring pre-estimated SVBRDF maps to a larger material sample. Using PatchMatch texture synthesis, Melendez *et al.* [Mel+12] transfer displacement and albedo maps from small samples of the materials. Their method is limited to daylight illumination and materials present in façades. By means of look-up-tables, and using surface normals and specular as guidance, Riviere *et al.* [RPG16] transfer surface reflectance captured with controlled LCD lighting to a material sample observed under natural lighting. Recently, Deschaintre *et al.* [Des+19] fine-tune a network trained to estimate SVBRDFs [Des+18], to work on larger material samples taking a guidance image as input. This approach is limited

to transfer a pre-defined set of property maps while our method can transfer any kind. The strategy of using multiple images of the material under different illuminations as input data is not new. Li *et al.* [Li+17a] and Ye *et al.* [Ye+18] utilize a self-augmentation strategy to make the estimation of the SVBRDF more robust to unknown environment illumination. We are inspired by these approaches to increase robustness in the model predictions.

Figure 2.3: Overview of the method. We learn a mapping between the photometric response I_L of the material and a visual property map ω . We make this mapping robust to affine transformations by means of a particular policy of data augmentation used for training. We learn one model M per material and visual attribute, which allows us to robustly evaluate the performance of the method under several transformations of the guidance images X . At evaluation time, M can have any size. The training of M per visual attribute w takes less than a minute.



2.3 Problem Formulation

Our goal is to transfer a D -dimensional spatially-varying *visual attribute* w of a material (for example, estimated locally at high resolution) to a larger sample of it.

We formulate the problem with an image-to-image translation approach. For training, our method takes as input: a *photometric* dataset I_L , and a *visual attribute* ω . The *photometric* dataset consisting of a number of RGB planar images of a material, $I_L = \{I_l | I_l \in \mathbb{R}^{n \times m \times 3}\}$, $|I_L| \geq 1$, illuminated with different light sources $l \in L$, of $n \times m$ pixels. This kind of images can be either captured with specific devices [Nam+16; MWK17; Alc+19], or synthetically rendered given an inverse material estimation pipeline [Guo+20b; DDB20]. The visual attribute being a spatially-varying map $\omega \in \mathbb{R}^{n \times m \times D}$ of any kind, and dimensions D , that maintains pixel-wise correspondence with the photometric input images I_L . Figure 2.3 and Figure 2.5 show examples of these images for three different visual attributes: a stylization, a segmentation, and a normal map.

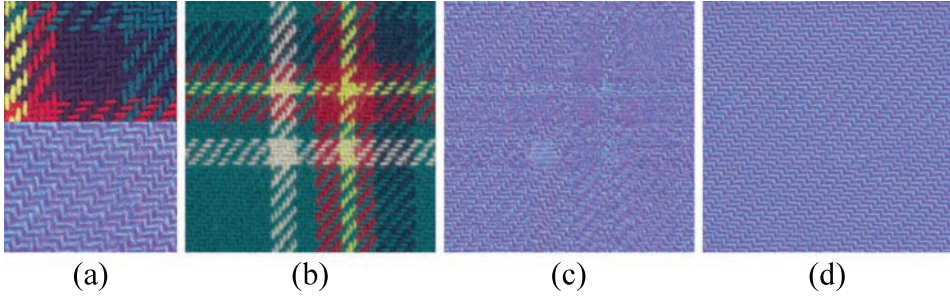
Given this data for a single material, and a strategy of patch-based training, we learn a function M that can be applied to a new *guidance* image of the input material (or a similar one) $X \in \mathbb{R}^{N \times M \times 3}$ of any size $N \times M$, to get its corresponding visual attribute $\Omega \in \mathbb{R}^{N \times M \times D}$.

$$\mathcal{M}: X \rightarrow \Omega \quad (2.1)$$

$$\text{s.t. } i_l \xrightarrow{\mathcal{M}} \omega, \forall i_l \in \mathcal{I}_L \quad (2.2)$$

At evaluation time, the guidance image X might contain different colors, scales, illumination, or affine distortions than the images used for training. Figure 2.3 shows an overview of the training and evaluation processes. In Section 2.6, we evaluate the conditions of the input guidance image upon which the method provides robust estimations.

Figure 2.4: The color augmentation policy takes advantage of the material structural regularities to make the transfer robust to different albedos. (a) A diffuse image of the portion of the material used for training, and its corresponding normal map. (b) Input guidance image. (c) Transferred normals without the color augmentation policy. (d) Transferred normals using color augmentation. Note that the training data does not include images containing the white yarn.



2.4 Learning Framework

In this section, we describe the patch-based and data augmentation strategies used for training, the neural network design and loss functions, and the implementation details.

2.4.1 Patch-based Training

Each training step takes as input a pair of corresponding image patches taken from the photometric input I_L , and the visual attribute ω . Using this data alone already provides a good starting point for generalizing to unseen illumination setups. However, it is not sufficient in scenarios in which the guidance image contains variations due to image noise, a different scale, or any other affine distortion. In order to make the transfer invariant to these transformations, the network needs to be trained with the appropriate data. Data augmentation strategies are essential for reducing the amount of necessary data for training

[SK19; Kar+20a], however, random strategies not taking into account the particular domain might degrade the quality of the prediction. We therefore follow a pre-defined data augmentation policy T (illustrated in Figure 2.3), where random operations are performed sequentially.

Color Augmentation Even if the microstructure of the material is homogeneous and can be measured using only a small patch of it, it may not be possible for the network to estimate its visual attributes for parts of the material which contain previously unseen colors. We correct this by randomly permuting the color channels of the photometric input (see Figure 2.4). As we show in Section 2.7, this data augmentation policy also helps make the model generalize the transfer to similar materials, by learning features that are more related to the structure of the material than to its color. Similar operations have been recently proposed for finding robust visual representations on self-supervised settings [Che+20b]. Only the photometric input is subject to this transformation.

Affine Transforms In order to allow for material editing applications in real images, the model should generalize to images taken under camera perspectives and geometry distortions different than those present in the fronto-planar images used for training. Many of such texture irregularities can be defined as an affine transform: translation, rotation, shear, or scaling. As CNNs are shift-invariant by design [Kau17], we propose to augment our datasets with random transformations for scalings, rotations, and shears. Those transforms can be efficiently performed on images through matrix multiplications. However, some visual property maps need to be treated specially, as the spatial transform might have a different behavior in 3D vector space, e.g. normals or tangents. In those maps, each pixel is a representation of a 3D vector. As such, we also perform the rotations and shear operations to these maps in 3D space, by multiplying each normal vector by the same affine transformation matrix applied to the 2D image. We perform the rotation around the Z axis, thus, assuming the camera sensor is parallel to the object plane.

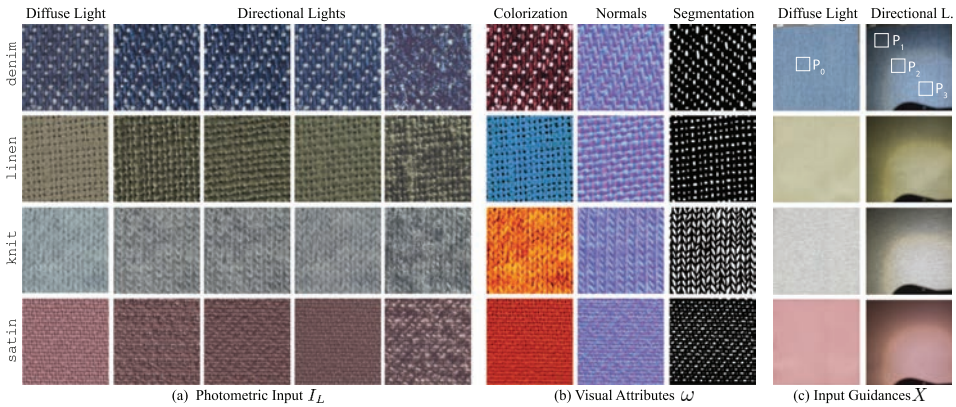
Cropping Inspired by recent work on patch-based learning [Tex+20b; Par+20], during training, the network receives small patches of each input pair. Those patches are randomly cropped from the randomly augmented images, so the network receives a considerable amount of variations of the same material, thus making generalization possible.

2.4.2 Network Design

We follow a uni-modal image-to-image translation learning strategy, assuming there is only one correct mapping from input to output image. This approach is reasonable for the kind of transfers we test in this work. However, it might

fail for ambiguous cases where there are multiple suitable outputs for the same input, for which multi-modal approaches [Zhu+17b] or GANs are more advisable, albeit harder to train. Specifically, our model M is a shallow U-Net network [RFB15] with 4 blocks of layers, containing a small number of trainable parameters, inspired on few-shot learning strategies [Wan+20d; Wan+19; Liu+19b]. This type of fully convolutional architecture is of common usage in image-space regression problems, due to its capability of efficiently learning patterns at different levels of abstraction thanks to its multi-scale design. Skip connections are added to enhance local details [RFB15; Iso+17]. The small number of parameters allows for faster training and inference, as well as reduced memory usage.

Figure 2.5: An overview of our evaluation dataset. (a) Five example images of our high resolution captured data illuminated with diffuse light and four directional light sources. (b) Examples of some of our *visual property maps*: colorization, normals and yarn segmentation. (c) Evaluation images taken under diffuse and directional illumination sources. *Denim*, (c) contains labels for several patches used for the evaluation in Section 2.6. Those materials show different properties that may prove challenging for our visual attribute transfer task. Our *denim* and *knit* fabrics show diverse color variations (different colored dyed yarns and stochastic albedo, respectively), *linen* shows strong geometric variations and *satin* shows anisotropic optical behavior.



Note that, as opposed to previous work [DDB20] on material transfer, which is initialized from a pre-trained network, but inspired by single-sample image synthesis methods [Zho+18; SDM19], we train one network per material and visual property. This strategy, although increasing training time, is key for the following reasons: First, it allows us to better understand the generalization capabilities obtained through our data augmentation policies. Second, it guarantees predictability of the trained model as every feature learned by the network is specific for each material and visual property pairs. Finally, it removes potential problems of a biased dataset as there is no cross-material

or cross-domain learning. If the input dataset does not contain enough variations of the material to represent the whole material, the model will fail in areas with unseen patterns. In those cases, having a pre-trained network may help as a material prior, as in [DDB20]. However, our method obtains comparable results to pre-trained methods, with a smaller computational footprint and with the additional flexibility of not needing an expensive dataset and large models. In practice, training a single network per material and attribute is not problematic as this process takes less than a minute.

Loss Function Choosing the appropriate loss function for a learning framework highly depends on the problem. For example, some methods [Iso+17] combine a per-pixel L_1 metric with adversarial [Goo+14a] losses, as the latter allows for better semantic mappings in multi-modal learning scenarios, whilst the former allows for improved predictability. Texler *et al.* [Tex+20b] further includes a perceptual loss [Tex+20a], while using a render loss is common in methods that estimate material parameters from photos [Des+18; DDB20]. In our method, we show that using a L_1 loss for training is enough for learning accurate and predictable mappings in our regression task, and binary cross-entropy loss for semantic segmentation, following standard practice in image segmentation [RFB15]. We found that the L_2 loss function yields overly-smooth outputs, and perceptual loss functions like $LPIPS$ [Zha+18] are more prone to artifacts than pixel-wise ℓ_{norm} metrics. We include an ablation study of the impact of the loss function in the supplementary material of this chapter.

2.4.3 Implementation details

We use PyTorch [Pas+19] as the learning framework, Adam [KB15] for optimization, a learning rate of 0.002, and a batch size of 16. The training images are randomly augmented using uniform distributions by the following operations, in order: First, the photometric input is subject to the color augmentation policy. Then, both inputs and targets are randomly rotated by an angle in the $[-90, 90]$ range, randomly sheared by an angle in the $[-45, 45]$ range, and randomly rescaled in the $[0.5, 2]$ range of scale factors. Then, patches of 128×128 pixels are randomly cropped during training to generate a large dataset of images.

All inputs are always standardized using their own mean and variance. Each M is trained for 1000 iterations, which takes around 1 minute on a single Nvidia 1080Ti GPU. Due to the fully convolutional nature of our models and their reduced number of trainable parameters, the guidance images X used for evaluation can be of arbitrary dimensions. We used images of up to 5000×5000 pixels for which evaluation takes around 150ms. We refer the reader to

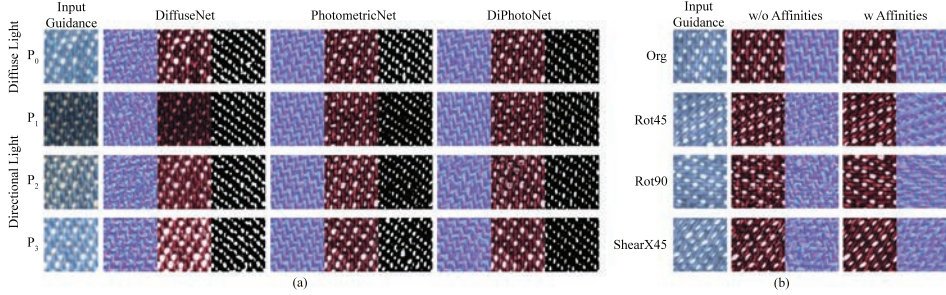
the supplementary material in this chapter for a comprehensive description and a diagram of this model, as well as further implementation details.

2.5 Dataset and Metrics

2.5.1 Dual-Resolution Captured Data

Our method is agnostic to the capture setup [Nam+16; MWK17; Alc+19], and it may work for any kind of input data (e.g. BTFs [LM01; Rai+19]) as long as the photometric images are pixel-wise aligned with each other and to the visual attribute. This dataset could also be created synthetically by rendering the outputs of any SVBRDF estimation method [DDB20]. In Section 2.7, we show results of our method using these acquisition pipelines.

Figure 2.6: Qualitative results of our models under different datasets and data augmentation configurations, for different inputs of the *denim* material. On the left (a), we show the results of our networks trained using different dataset configurations, under a guidance image taken with diffuse lighting (p_0), and three crops of a guidance image illuminated with a directional light (p_1, p_2, p_3) not present in the training set. Please refer to Figure 2.5 for the position of these crops on the larger guidance images. On the right (b), we show the results of our *photometricNet*, under different geometric distortions (rotations and shears) performed to its guidances images, taken under diffuse illumination.



However, for the purpose of this evaluation, we choose to work with real data. The main reason is that the data obtained with material capture devices poses extra challenges that are difficult to reproduce with render engines. For example, material irregularities, complex optical behavior, or distortions and color shifts introduced by the optical system that might cause the models to produce inaccurate estimations.

We create a dataset containing images of the same material taken with two imaging camera systems: a high-resolution camera that allows us to take pictures of 0.7×0.9 cm, with a resolution of 367×490 pixels and a macroscopic camera which provides images of 11×11 cm, with a resolution of 4800×4800 pixels. In

terms of illumination, our setup has 27 different collimated light sources uniformly distributed across the hemisphere as well as diffuse illumination. We build a dataset of four different textile materials, whose complex optical behavior due to anisotropy, transmittance, directionality and microstructure [CLA19] turns them particularly challenging for synthesis and editing operations [Kam+16; Rai+19]. For training, we capture one image for each light source, making a total of 28 different images (Figure 2.5 (a)). For the evaluation set, we take one *guidance* image with diffuse illumination and another *guidance* image with a directional light source (Figure 2.5 (c)). The visual attributes (Figure 2.5 (b)) are generated automatically using photometric stereo [Ike81] in the case of normal maps and manually by artists in the cases of colorizations and segmentation masks.

2.5.2 Attribute-specific Metrics for Evaluation

For evaluating our models, we choose domain-specific distance metrics different from those they were trained with to better understand their generalization capabilities [TOB15]. For normal maps, we compute the cosine distance between ground truth and estimated maps, as it accounts for the geometric space in which normals lie. In the case of image segmentation, we evaluate the results using the *Jaccard similarity coefficient* (IoU) [Zho+19], which is well-suited for sparse segmentation tasks. Finally, we use the state-of-the-art metric *Learned Perceptual Image Patch Similarity* (LPIPS) to evaluate the quality of the colorizations, as it has been shown to outperform L_2 norm for visual perception tasks [Zha+18].

2.6 Evaluation

We evaluate our method in different settings. First, we assess the type and amount of photometric input data necessary for the model to generalize to different illumination conditions and image distortions. Then, we test our data augmentation strategy for affine transformations.

2.6.1 Invariance to Input Illuminations and Distortions

In this set of experiments, we aim to evaluate if the model produces the same output after changing the training data and input guidance images. To this end, we measure both the impact of different illuminations and sizes of the photometric dataset, as well as variations of the input guidance images.

Photometric Input

In this experiment, we study the type of photometric input that makes the method invariant to different illumination conditions of the guidance image. We

compare three models trained with different datasets: *diffuseNet*, which uses a single image illuminated with diffuse lighting; *photometricNet*, which takes as input 27 directional lights; and *diphotoNet*, which uses all the 28 sources. For evaluation, we use the macroscopic camera which captures a larger sample of the material at a lower resolution. We take two test guidance images with diffuse and directional lighting (Figure 2.5 (c)). In this experiment, all images are aligned, therefore, we follow a limited policy of data augmentation from Section 2.4.1, applying only color augmentation, rescaling, and crops, and leaving out rotations and shears for in-the-wild scenarios.

Figure 2.6 shows qualitative results of our method for a selection of patches for the *denim* material. Figure 2.6 (a) shows that the best accuracy is in general obtained with *diphotoNet*, i.e. training with both directional and diffuse illuminations. This is reasonable, as the model is trained using the same illuminations used during test time. Conversely, *diffuseNet*, a network only trained using diffuse lighting is less capable of generalizing under any kind of illumination source. This suggests that following a photometric approach for training material synthesis models allows for better generalization capabilities. Finally, the results for *photometricNet*, which is trained only with directional lights shows similar accuracy as *diphotoNet* while proving generalization to unseen illuminations. For all the results shown in the chapter, we have chosen *photometricNet* as our model. This has an additional advantage from the usability perspective: it is relatively easy and cost-effective to build a capture setup (such as the flash light from a smartphone) or generate renders with directional lights, while it is considerably harder to recreate both types of illumination consistently.

Dataset Size Influence

Our goal in this experiment is to understand how many images [NJR15] taken under directional lights are needed in order to obtain the desired invariance to illumination.

We train different models using reduced versions of our full 27-image dataset, and compare their results to those of the *photometricNet* trained on the full directional dataset. More precisely, we train networks using 1, 3, 9 and 18 directional lights for each material and application in our dataset following the same reduced data augmentation policy described in the previous experiment. Instead of randomly selecting light sources around the hemisphere, we perform a more sensible light source sampling. Specifically, for each reduced dataset, there is at least one light that is as close as possible to the normal of the surface on which the material lies, thus giving more importance to frontal angles. We extend these selected lights to 3 for the reduced datasets with more than three lights. The rest of lighting sources are uniformly sampled around the hemisphere, up to a zenith angle of 70 degrees. The supplementary material contains a diagram of the distribution of lights in the hemisphere. Results are shown in Figure 2.7, where we see

that adding more lights to the dataset monotonically increases the generalization of every model. However, adding lights consistently shows diminishing returns, which suggests that a capture setup with around nine lights might be enough for relatively accurate estimations. Similar findings are reported in multi-image SVBRDF estimation methods [Guo+20b; Des+19].

Figure 2.7: Error of the reduced *photometricNets* on our ground truth guidance image (taken with diffuse illumination, not present in the training dataset) for each material in the dataset and the three visual attributes and corresponding error metrics, which are close to the normal of the surface in which the material lies. The width of the lines indicates the standard variation across 5 repeated experiments, where different light directions were randomly chosen to form the training datasets. Please refer to the supplementary material for the position of each light source.

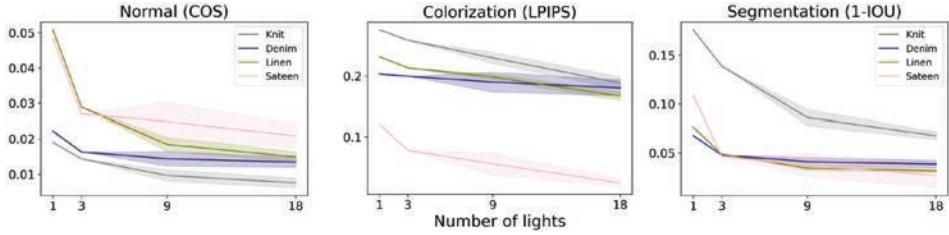


Image Degradations

As discussed in Section 2.5.1, real captured images may be subject to distortions, shifts and noise introduced by the optical capture system. To fully understand the robustness of our models with respect to these types of imperfections, we synthetically modify the saturation, contrast and noise present in the X guidance images. As shown in Figure 2.8, M is robust to these types of degradations, even when a fair amount of details are lost in X . These images were not part of the training dataset.

2.6.2 Equivariance to Affine Transforms

Our goal in this experiment is to analyze the equivariance of our model to affine distortions of the guidance images. A function f is said to be *equivariant* with respect to a transformation T if $f(T(x)) = T(f(x))$. Inspired by [Ben+20], we measure the equivariance of our model M with respect to different affine transforms T performed to an input guidance image X , by computing their difference using the corresponding metric defined in Section 2.5.2 $d(M(T(X)), T(M(X)))$. To achieve this, we extend the augmentation policy in the previous experiments by adding random shears and rotations to the training process. We train the *photometricNet* with and without random shears and rotations for every visual attribute in our dataset and on the *denim* material. In the case of the normals, as described in

Section 2.4.1, we also perform the shears and rotations in the geometric space in which normals lie.

We measure their robustness with respect to three different transformations T : rescalings, rotations and shears. As before, we use a guidance image taken under diffuse lighting as input for these experiments and attribute-specific distance metrics. Figure 2.6 (b) shows that not adding those transforms generates visual artifacts for rotations and shears. Interestingly, without affine augmentations, the models hallucinates vertical yarns, as it is the only type of data that it has seen as input. Figure 2.9 shows quantitative metrics for the range of transformations T in which we evaluated the models. Augmenting the training dataset with both shears and rotations generally provides the best results. Furthermore, this enhanced data augmentation policy improves the robustness of the models with respect to the scale of their inputs. Notably, applying random rotations during training appears to have a larger impact than random shears on the robustness of the models. This finding suggests that applying blind policies of data augmentation [ASE17; SK19; San+19; Che+20b] may not be an optimal strategy for some applications like ours, as the networks may not learn relevant features, or overfit to noise present in the training dataset. Finally, it is worth noting that every network has the same number of parameters, and is trained for the same number of iterations, which shows that the generalization capabilities of the models can be increased at no extra parameter cost.

Figure 2.8: Output of our model under different distorted inputs: a change of saturation, contrast, and Gaussian noise ($\sigma^2 = 255$). The number at the bottom is the cosine distance with respect to the original estimation. As we can see, the output is consistent with a very small error in all cases.

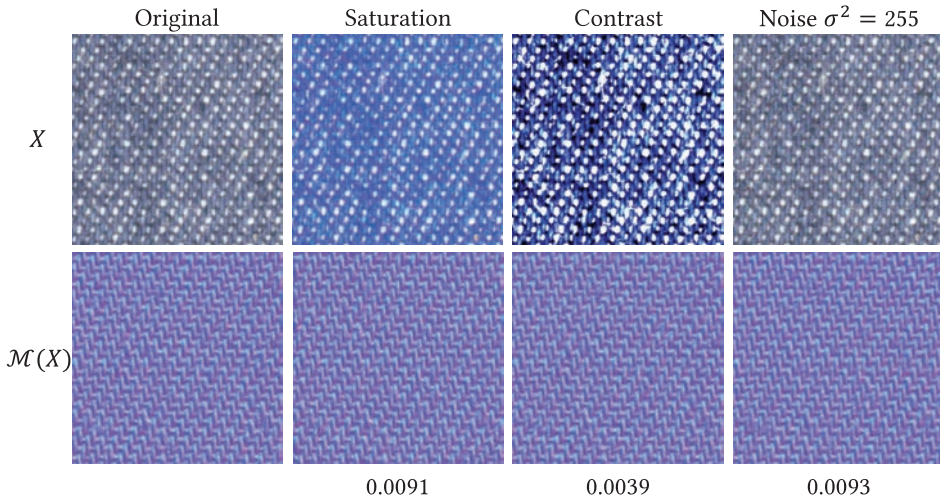
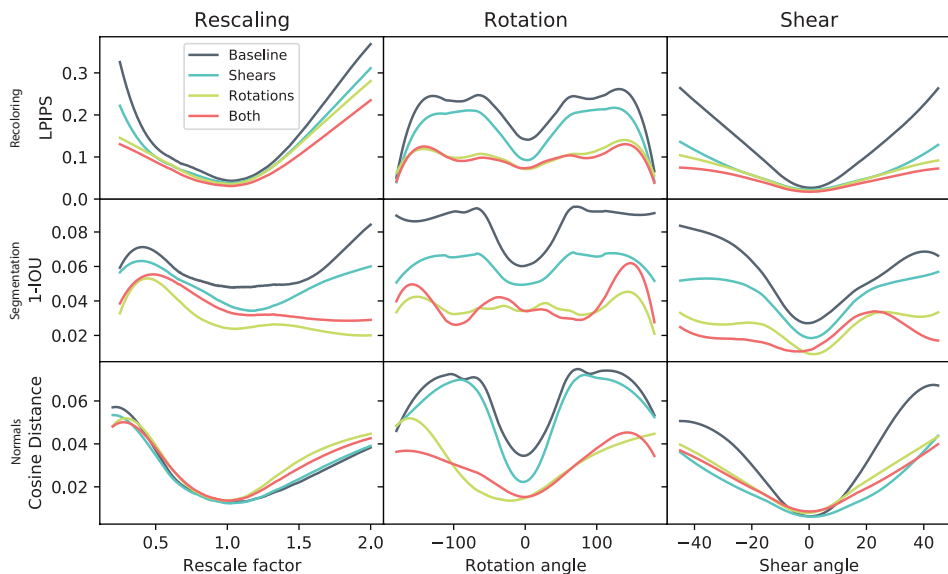


Figure 2.9: Quantitative evaluation of the robustness of our networks with respect to different transformations T (rescaling, rotation and shearing) of networks trained under different data augmentation policies for the *denim* material. From top to bottom, we show results on: recoloring, segmentation, and normals estimation. Lower is better for each metric



2.7 Results and Comparisons

In this section, we first compare our method with related approaches on image stylization and large scale SVBRDF material transfer. Then, we show the capabilities of our method to generalize to similar materials to those in their training set and present its limitations.

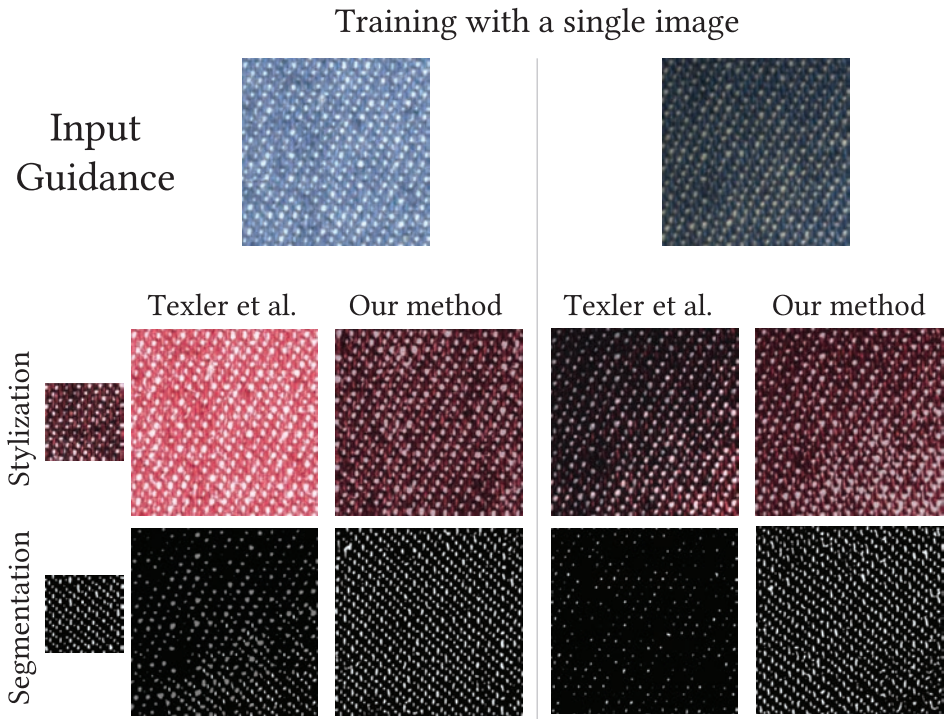
2.7.1 Interactive Stylizations

In the first category, the method of Texler *et al.* [Tex+20a] allows artists to interactively edit a few keyframes of a video and propagate that edition to the rest of the video. As ours, they formulate this transfer problem by training an encoder-decoder network using patch-based learning. But, in contrast to our work, they do not perform any data augmentation policy outside of random cropping. For a fair comparison with such method, we compare two setups: 1) using a single image as training input, and 2) using the photometric dataset. Results are shown in Figures 2.2 and 2.10. As Texler's method is not scale invariant, in both cases the training data provided for their model has the same scale as the images used for testing.

Our models have been trained with the full policy of data augmentation. In the first setup (Figure 2.10), we use the diffuse illumination and hence compare their output with our *diffuseNet* output. As shown, none of the methods provide high quality results but our model manages to provide closer estimations. In the second setup (Figure 2.2), we train their model with the photometric input. The best results are obtained with our method. Even though extending the input data using photometric cues has an impact on the quality of Texler’s results, the lack of a data augmentation policy makes the transfer fuzzier and noisier. Further, their combination of style, adversarial and pixel-wise losses fails to yield predictable mappings. These results confirm the importance of a comprehensive data-augmentation policy, such as the one we propose, when using neural networks for image processing tasks of this kind. In addition, our model is trained in less time with a smaller computational footprint (1 minute vs 5 minutes).

Figure 2.10: Comparison of our method with Texler *et al.* [Tex+20b] using a single diffuse image of the *denim* material as training data. The task is to transfer the two attributes shown (*Stylization* and *Segmentation*) to two different guidance images.

Even using a single image instead of a photometric dataset, we achieve higher quality mappings at a lower cost.



Another way of formulating this visual attribute transfer problem is through *image analogies*. Using our single diffuse image for input, we compare our approach with two methods as shown in Figure 2.11. First, the work of Liao *et al.* [Lia+17], which uses deep latent spaces as image descriptors; and the method of Benaim *et al.* [Ben+21] that trains single-image generative models to find bijective mappings between the *structure* of one image and the *style* of another. Our method qualitatively outperforms these methods with a fraction of the computational cost: 1 minute in our case, 10 minutes in [Her+01], 40 minutes in [Lia+17] and 10 hours in [Ben+21]. Once trained, our models can be used to evaluate any guidance image in real time for materials with similar microstructure. In contrast, image analogies methods require expensive optimizations for each guidance image. We refer the reader to the supplementary material for more comparisons with these methods.

Figure 2.12 shows additional results of material stylizations. In these examples, we used the method of Gatys *et al.* [GEB15b] to stylize a small patch of the material. Then, we trained a model using photometricNet and diffuseNet. As the guidance image we used a bigger image with diffuse illumination. Compared with naïve style transfer applied to the whole image, our approach provides detailed stylizations where the microstructure of the material is preserved. We further see that diffuseNet provides noisier results than photometricNet, probably due to the fact that the photometric cues help to preserve the local shading variations.

2.7.2 Creation of Large Scale Digital Material Assets

Our method can be used to propagate SVBRDFs estimated locally in a small area of the material to larger samples. Figure 2.13 illustrates for a diverse set of materials that we can propagate albedo and normals estimated at high resolution in a small area of 0.7×0.9 cm to guidance images of 13×13 cm taken with a smartphone. Even though we train the albedo and normals models separately, which does not guarantee pixel-wise coherence between the estimated maps, the rendered images show realistic-looking materials, even with a diffuse material model. As opposed to the method of Deschaintre *et al.* [DDB20], which is trained to output directly Cook-Torrance [CT82] material layers, our method is agnostic to the parameters of the SVBRDF.

Figure 2.11: Comparison of image analogies approaches for input shown on the left, of the *denim* material. From left to right: Deep Image Analogies [Lia+17] and Structural Analogies [Ben+21]. Our results shows more accurate and predictable mappings, at less computational cost than the alternatives.

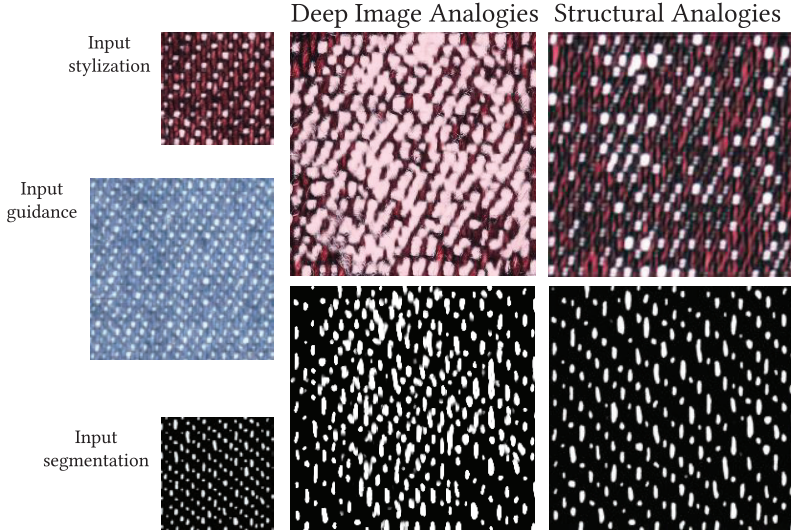


Figure 2.12: Our method allows for interactive material-aware visual attribute transfers. Using an off-the-shelf style transfer algorithm [GEB15b], we can transfer the style of one image I_{style} to the content of another $I_{content}$, obtaining a visual attribute ω . Training a M_ω to learn this relationship, we can find predictable style mappings, that we can transfer to guidance images X , obtaining style transfers $M_\omega(X)$. Learning this transfer is inexpensive and allows for interactive editions. Performing this transfer directly to the guidance image generates artifacts and not-predictable mappings.

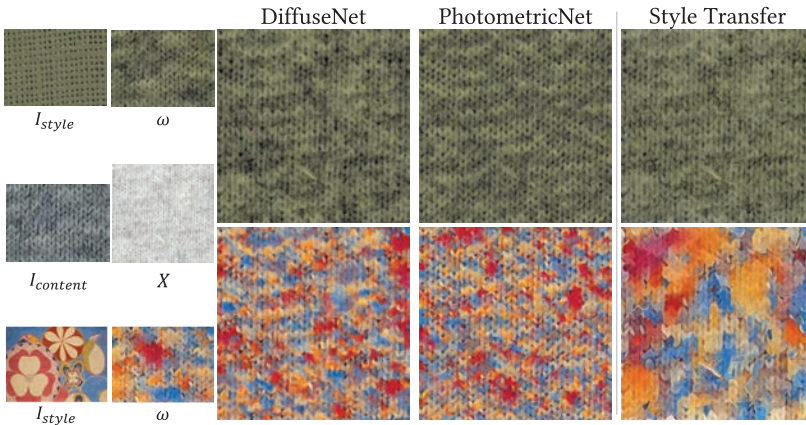


Figure 2.13: Results of our framework for material capture using a smartphone. Training two models, M_a and M_n with a photometric dataset, we obtain respectively albedo and normals from guidance images taken under uncontrolled conditions, which can be used by render engines. Here we have used Arnold [Geo+18] and a diffuse material model.

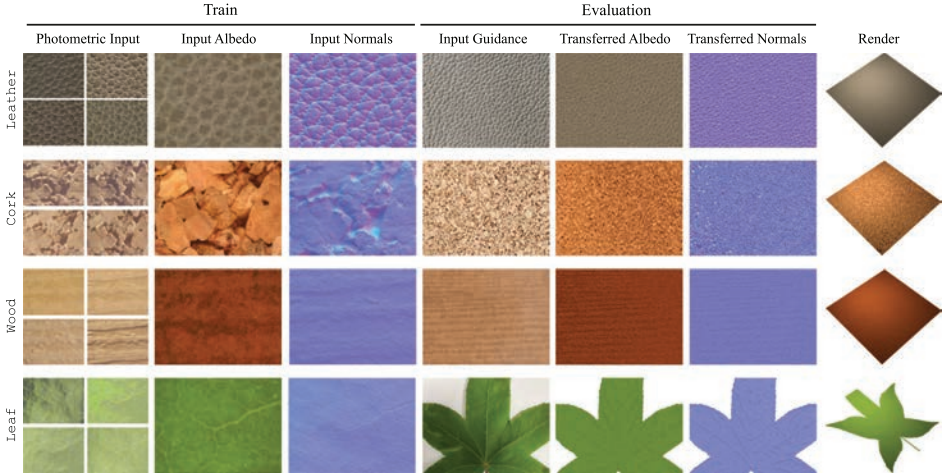
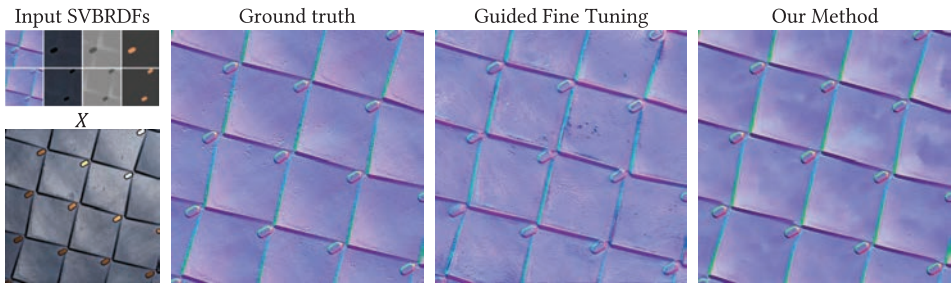
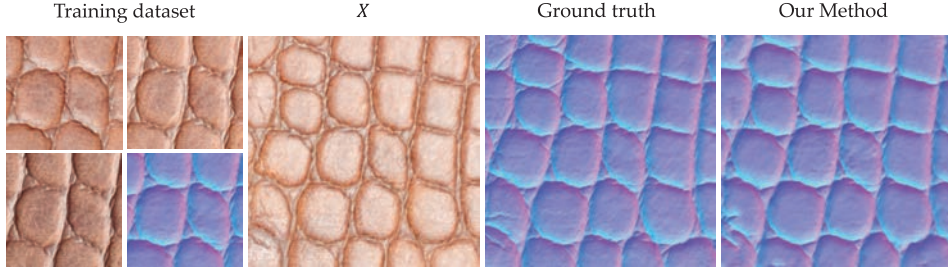


Figure 2.14: Comparison of our method with the Guided Fine Tuning, by Deschaintre *et al.* [DDB20]. Following their algorithm, we render the *Input SVBRDFs*, and train a photometricNet on those renders. As shown, our method can achieve higher quality normal maps, with fewer artifacts. Input SVBRDFs, X , ground truth and results from Guided Fine Tuning were obtained directly from [DDB20].



We assess the capabilities of our method on this setting using the same SVBRDF propagation scenario as proposed in [DDB20]. Using a small crop of a synthetic SVBRDF as input, we render 27 images using the same directional light position as we used in our real dataset, and train a photometricNet to estimate the surface normals from each of those renders. We then evaluate this model using a larger area of the material, illuminated under an unknown lighting position. In Table 2.1, we show a quantitative comparison

Figure 2.15: Our method can work with real BTF captured data. Training a photometricNet on a crop of the material (summarized in *Training Dataset*), to output surface normals, we can estimate surface normals on larger areas of the material, even under novel viewing positions.



with [DDB20], under different image quality metrics. As shown, our method achieves better scores on pixel-wise metrics, whilst [DDB20] achieves better deep perceptual scores, as in the LPIPS metric [Zha+18]. This might be related to the design of our loss function: we directly minimize pixel-wise differences, while [DDB20] is optimized using a render-aware loss. Qualitatively, as shown in Figure 2.14, our method obtains comparable quality mappings, with fewer artifacts. Despite capturing SVBRDF or arbitrary materials is not the goal of our method, we include in the supplementary comparisons with a direct SVBRDF acquisition method [Gao+19] using our own real dataset.

Table 2.1: Quantitative comparison with [DDB20], on the studied materials and different performance metrics. As shown, our method provides better pixel-wise accuracy than [DDB20], while their method obtains better perceptual scores.

Material ID	SSIM [Wan+04] $\uparrow$		PSNR $\uparrow$		MSE $\downarrow$		LPIPS [Zha+ 18] $\downarrow$	
	[DDB20]	Ours	[DDB20]	Ours	[DDB20]	Ours	[DDB20]	Ours
560	0,905	0,925	31,190	32,310	0,002	0,002	0,244	0,221
1581	0,645	0,675	29,740	29,920	0,006	0,005	0,398	0,446
1684	0,654	0,746	29,711	30,410	0,007	0,003	0,196	0,251
2111	0,770	0,783	32,250	32,881	0,002	0,002	0,397	0,383
Average	0,744	0,782	30,723	31,380	0,004	0,003	0,309	0,325

A similar setting can be used to propagate attributes leveraging BTF measurements as training data. Figure 2.15 shows an example using a BTF from [WGK14]. Using Photometric Stereo [Ike81], we compute their surface normals and train a photometricNet on a central crop of the BTF, using the captures at a camera position of ($\phi = 0^\circ$, $\theta = 0^\circ$). We then evaluate this model using the full

material surface, and a novel camera position, of ($\phi = 15^\circ$, $\theta = 11^\circ$). As shown, our model is capable of working with captured BTF data.

2.7.3 Generalization to Similar Materials

In previous experiments, we have shown the performance of our models when the guidance image corresponds to the material used for training. In this experiment, we show that our models, despite being trained only on a set of captures of a single material, generalize to materials of the same category. Figure 2.16 shows some examples for models trained using our dataset (Figure 2.5), taking input guidance images of different albedos and scales. The transfer works thanks to the network design and data augmentation strategy that is designed to use microstructure details as guidance. Our method could thus be used to transfer visual attributes for a diverse set of materials by simply training one model using a single but representative material of each category.

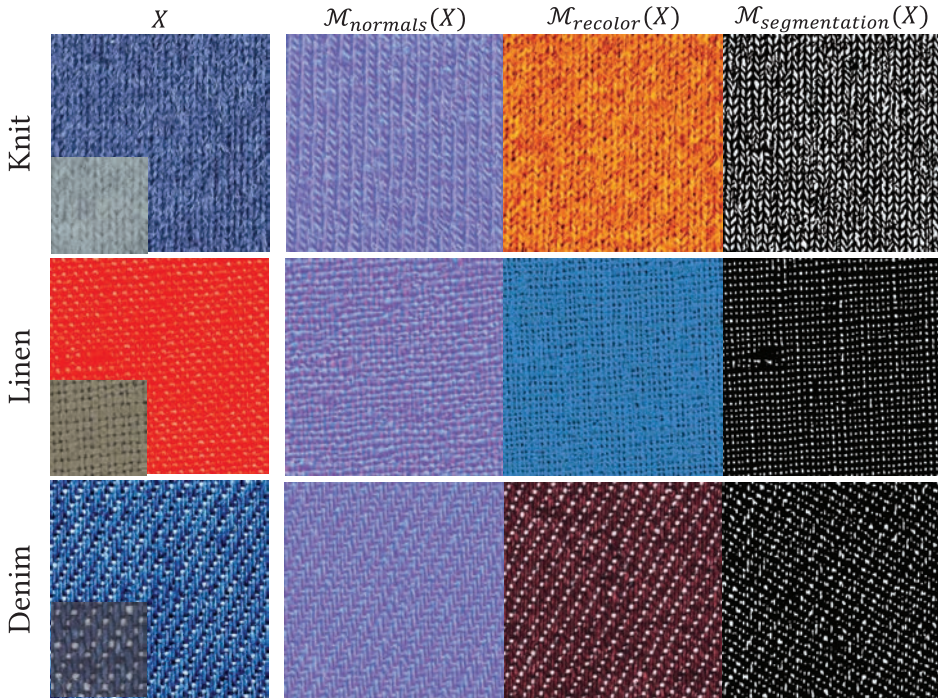
2.7.4 Limitations

Our models are not guaranteed to provide high-quality results outside the range of input data and data augmentation policies we train them on. This limitation is common to all learning-based approaches. It is unlikely that the framework is capable of generalizing to resolutions higher to those of the training data; or down-sampled images in which the texture details are not recognizable. Furthermore, the type of transformations we apply to the training data may not represent all the possible geometric variations, or non-linear warpings that materials are subject to in the real world. For materials which exhibit a strong variation in their microstructure, which cannot be fully captured using a single photometric dataset, our patch-based approach will likely fail to generalize to the full heterogeneity. Figure 2.17 shows two examples of failure cases for which the test data is not included in the training data and where the affine and illumination transformations are outside the suitable range.

2.8 Conclusions and Future Work

In this chapter, we have proposed a neural visual attribute transfer framework capable of transferring, for a given material, many types of visual property maps to images of unseen patches of the same -or similar- material taken under different illumination, capture setup, and affine distortion. To our knowledge, the proposed framework is the first method capable of leveraging the optical behavior of the material to this purpose by being trained using a photometric approach. Such an approach, besides the illumination-invariance we have shown, helps

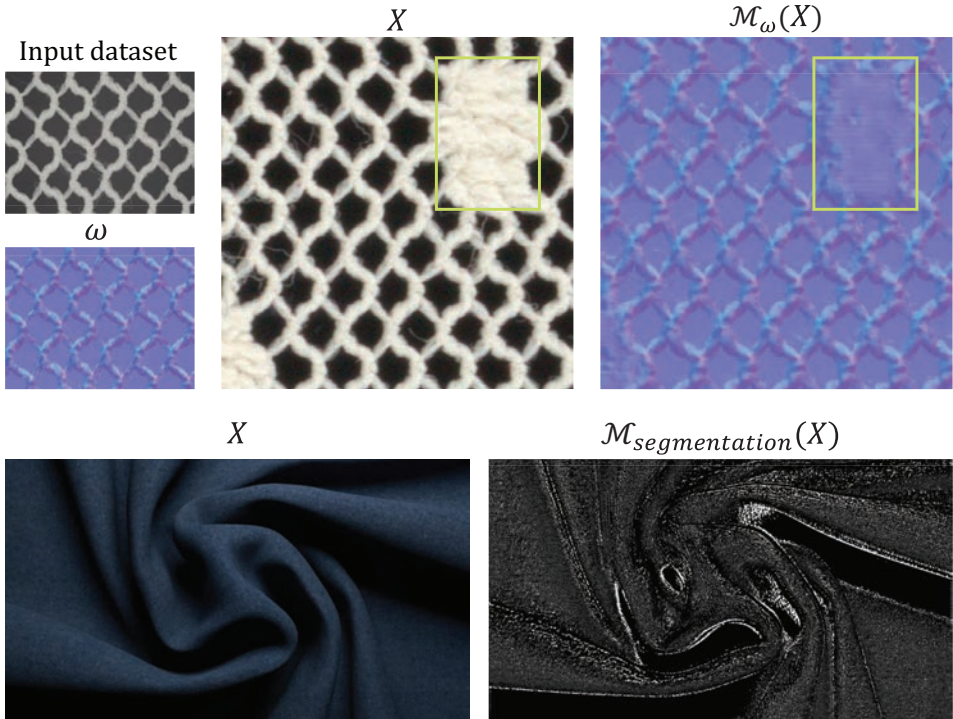
Figure 2.16: Generalization capabilities of our method when evaluated on materials similar to those in their training dataset. On the top row, we show the outputs of the model trained on the *knit* in our dataset (Figure 2.5), and evaluated on a different guidance images. We also show examples on linen and denim fabrics, with different conditions of saturation, blur, scale, and illumination. Even in very challenging cases, where the structure of the material is barely visible, as in the overly-saturated red linen or the noisy blue knit fabrics, our photometricNets can yield plausible results. The insets represent the training dataset by each model, as represented in Figure 2.5.



the neural network learn better mappings between visual domains, finding a physically-based representation of the material. Further, we have presented a comprehensive policy of data augmentation which outperforms previous work on visual attribute transfer given a single image of the material.

We have shown that our method can be used to transfer any kind of visual attribute estimated locally to larger material samples. Further, we have demonstrated that our models, although trained on a single material, generalize to materials of the same category. We think our findings will inspire future work showing that smart training strategies might alleviate the need for massive datasets. Our method could be extended in several ways. The need for obtaining high-resolution captures taken under

Figure 2.17: Failure cases of our method. As shown in the first row, if the input dataset does not represent the heterogeneity present in the guidance X , the model fails to yield compelling results on unseen structures of the material, as shown on the green box. On the second row, we show a guidance image of the *denim* material X which exhibits strong geometric and illumination variations, outside of the range in which we train M with. As such, the model shows a poor performance on the segmentation task.



different illuminations may be reduced by generating rendering images through recent advances in inverse material acquisition [Don19]. Further training with multiple patches may help to cope with material heterogeneity [Tex+20b]. Similarly, our findings suggest that extending the data augmentation policy to include 3D deformations will likely improve the accuracy. Beyond the generation of large scale digital assets for rendering, our method may have potential in other visual computing applications that require a low level understanding of the properties of the materials in real scenes. For example, the yarn segmentation application shown in the chapter might be suitable as input to shape from texture applications. Specific visual attributes might be useful to identify or highlight defects for image forensics problems, or to enhance different features in real-time or AR applications.

2.A Additional Implementation Details

Network design

Our model, summarized in Figure 2.18, is a standard U-Net [RFB15] encoder-decoder network with 4 blocks of layers and skip connections. U-Nets are widely used in pixel-wise regression and image-to-image translation problems, due to their efficient design, capable of exploiting texture and semantic patterns at different levels of abstraction, due to their multi-scale design. Skip connections are known to significantly enhance the quality of deep image-to-image translation models [Iso+17]. Each block is comprised of a tuple of convolutional layers with kernel sizes of 3×3 , followed by a batch normalization operation [IS15] and a ReLU [NH10] non-linearity. The number of convolutional filters in each layer is shown in Figure 2.18. The last layer of the network uses 1×1 convolutional filters to transform the feature vectors into the desired number of classes (e.g 3 output channels for RGB regression tasks, as in normal map estimation or recoloring; or 1 output channel for the semantic segmentation problem). The total number of trainable parameters depends on the number of property maps that the network should jointly learn, but are around 483000 for all the problems we showcased in this work. Trainable weights are initialized randomly, by sampling a $N(\mu = 0, \sigma = 0.02)$. The output of each block in the encoder part of the network is max-pooled to halve its spatial resolution. Upsampling in the decoder side of the network is done through up-convolutions. For image segmentation problems, we add a Sigmoid non-linearity to the output of the network. Due to the reduced number of trainable parameters, models take only 1.94Mbs, allowing for fast training and evaluation. On Table 2.2, we show a quantitative study on the influence of the depth of the model in its accuracy, using the 1684 material from [Des+18] as the training dataset. A visualization of these results is included in Figure 2.19, showing that deeper models yield overly-smooth results for this few-shot learning problem.

Table 2.2: Quantitative comparison between different number of blocks of layers in our model. Every model we use throughout this work has 4 blocks of layers, as it provides the best trade-off between computational cost and precision. Deeper models fail to generalize properly in this few-shot learning scenario. We use a color code to highlight **best** and **worst** cases

Depth	SSIM [Wan+04] $\uparrow$	PSNR $\uparrow$	MSE $\downarrow$	LPIPS [Zha+18] $\downarrow$	Size (Mbs) $\downarrow$	Train Time (s) $\downarrow$
3	0.71	29.83	0.0052	0.37	0.49	51
4	0.74	30.41	0.0031	0.25	1.94	58
5	0.72	30.37	0.0043	0.33	7.63	164
6	0.72	30.87	0.0041	0.31	30.49	283

Figure 2.18: An overview of our deep learning architecture M_{ω_i} , which is a shallow U-net network [RFB15]. Below each block of layers, we show the number of trainable convolutional kernels, as well as the spatial resolution spanned by each block, with respect to the input resolution D . We use the normal map estimation problem for a denim fabric for visualization purposes.

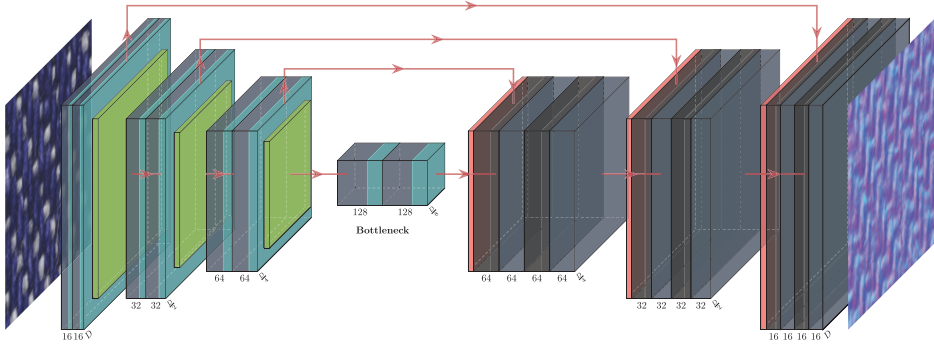
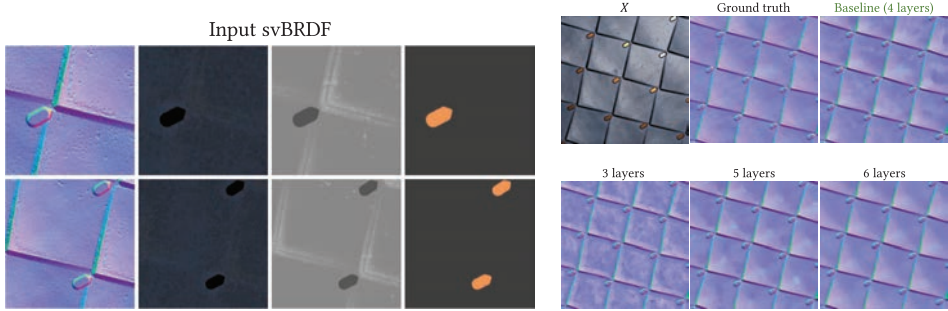


Figure 2.19: Impact of the depth of the model on its accuracy, using a synthetic dataset. As shown, our baseline outperforms deeper and shallower configurations.



Training details

We use PyTorch [Pas+19] as the learning framework, Adam [KB15] for optimization, a learning rate of $\alpha = 0.002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and a batch size of 16. The training images are randomly augmented using a uniform distribution by the following operations, in order: randomly rotated by an angle in the $[-90, 90]$ range, randomly sheared by an angle in the $[-45, 45]$ range, and randomly rescaled in the $[0.5, 2]$ range of scale factors. Their color channels are randomly reordered and patches of 128×128 pixels are randomly cropped during training, to generate a large dataset of images. The rescaling factors and rotation angles are randomly chosen for each image in each batch using a uniform distribution on the mentioned ranges. All these data augmentation

operations are also applied to the target visual attributes, except for the color channel reordering, which is only applied to the input images. Images and visual attributes are always standardized using their own mean and variance. All the data augmentation operations are performed natively in GPU, which allows for faster training. To further reduce the computational cost and training times of our models, we train our models using automatic mixed precision training [Mic+18], which reduces the memory footprint of the models in GPU, while allowing for faster computation of certain operations, such as convolutions.

Each M_ω is trained independently for each material and application (no cross-domain training or fine-tuning) for 1000 iterations, which takes around 1 minute on a single 1080Ti GPU. Due to the fully convolutional nature of our models, and their reduced number of trainable parameters, the guidance images X used for evaluation can be of arbitrary resolutions. We evaluate the models using half-precision, which allows us to utilize guidance images of up to 5000×5000 pixels, for which evaluation takes around 150ms in GPU. Thanks to the reduced number of parameters, an efficient use of GPU-native data augmentation techniques and leveraging half precision training and evaluation, we are able to train and evaluate our photometricNets in less than a minute of total computational time.

Ablation Study of the Loss Function

Finally, in Figure 2.20, we show a comparison of the results of our models when trained on different loss functions. As shown, even though the differences are sometimes subtle, the L_1 consistently yields sharper estimations than the L_2 , whilst the perceptual loss function is more artifact-prone and thus less predictable. L_1 is a common loss function in many image regression tasks, as in image-to-image translation [Iso+17] or in image synthesis [Zho+18].

Additional Dataset Details

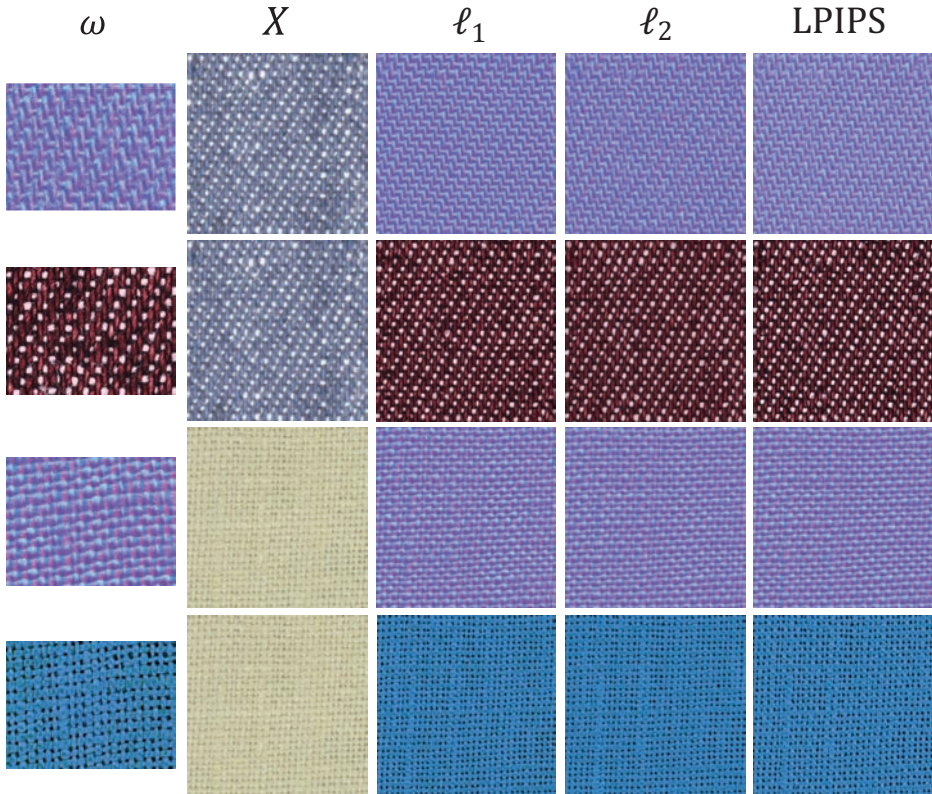
We show the positions of the lights in the photometric dataset in Figure 2.21.

2.B Additional Comparisons

Comparison with Image Analogies Methods

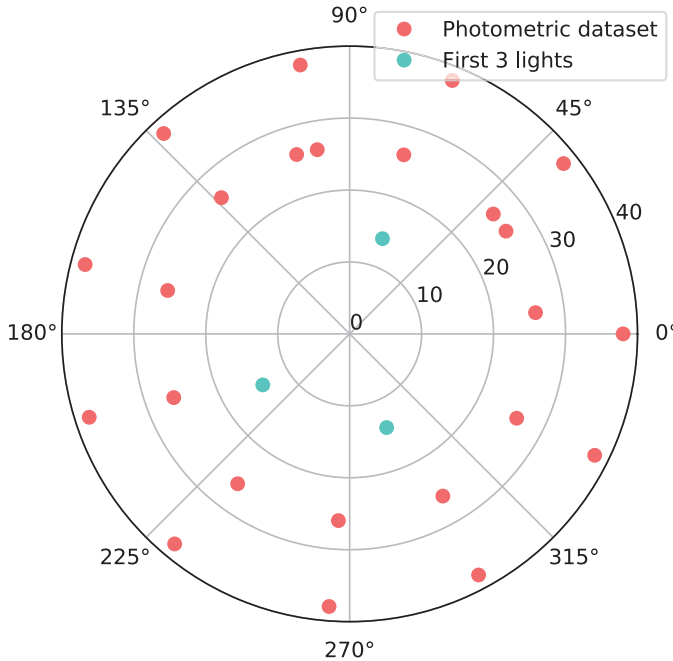
In Figure 2.22, we compare our method to different image analogies approaches. We show a comparison between [Her+01], [Lia+17], [Ben+21], and our method. The seminal work in image analogies [Her+01] fails in this visual attribute transfer problem, most likely due to the difference in scale between input and target images. For the comparisons with [Lia+17] and [Ben+21], we reduced

Figure 2.20: Impact of the choice of the loss function used to train the networks in the estimated maps at macroscale. On the left, input guidance X and visual attribute ω to be transferred. On their right, the result of M_ω when trained using L_p , L_2 and $LPIPS$ [Zha+18] loss functions. All the networks in this experiment were trained with the same configuration, only the loss function was modified.



the scale of the target images B to 256×256 pixels to make their methods computationally tractable. The methods in [Lia+17; Ben+21] never see the diffuse image A , as they generate an intermediate representation by themselves. While showing better results than [Her+01], methods which use deep learning for this task still fail to yield compelling or predictable results. The method in [Ben+21], which leverages structural information in the images to train a single-image generative model [SDM19], shows more plausible results, but with an expensive computational footprint (10 hours per network), compared to the 45 minutes in [Lia+17], 10 minutes in [Her+01] and less than 1 minute in our case. Because we assume pixel-wise semantic correspondence between input image and visual attributes, we can find mappings in interactive times, and which are significantly more predictable and of higher quality than these image analogies methods.

Figure 2.21: Relative position of the 27 directional light sources used to capture the photometric datasets with respect to the material sample, in degrees. In blue, we show the three lights that are always included in our reduced photometric datasets for the light sampling experiments. As shown, the lights cover a wide variety of azimuth and zenith angles, allowing for an accurate capture of the optical behavior of the materials.



Comparison with Texler et al.

In Figure 2.23, we show a comparison with different configurations of [Tex+20b]. Specifically, we train their method using a dataset with a single image using diffuse illumination (left) and the full photometric dataset (middle column). Their method is trained individually for each dataset and visual attribute, for around 5 minutes to guarantee convergence, as suggested in their paper. As [Tex+20b] assumes that the scale (pixels/cm) of the dataset and evaluation images are the same, we rescale our input dataset to match the scale of the evaluation images. We evaluate each method using three different images X , one with diffuse-light illumination (top row), another with diffuse-light illumination but with a geometric transformation (middle row), and an image captured with directional lighting (bottom row). As shown, by enhancing the method in [Tex+20b] with a photometric dataset, their model is able to generalize better to every image X in the evaluation dataset than its counterpart trained only using a single image. However, our method, specifically

designed for this illumination and geometry invariant visual attribute transfer outperforms their work in every type of input and visual attribute, whilst maintaining a lower computational overhead (1 minute of training in our case) and memory footprint (their trained network uses up 12 Mbs, while ours is 1.9 Mbs). Our model trained using a single illumination (diffuseNet) shows better results than their method trained using the photometric dataset, suggesting that our transfer method has more adequate inductive biases for the visual attribute transfer problem.

Comparison with deep SVBRDF acquisition

In Figure 2.24, we show a comparison with a recent deep SVBRDF acquisition method [Gao+19], which relies on a pre-trained autoencoder and a differentiable render engine for optimizing the initial estimation of the autoencoder. As shown, our method generates sharper and more coherent normal maps. PhotometricNet relies on a small sample of the same material for training, and is thus less sensitive to potential biases of the training dataset, with the cost of having reduced generalization capabilities compared to multi-material methods.

Figure 2.22: Comparison of photometricNet with related image analogies methods.

In order, we show a comparison between [Her+01], [Lia+17], [Ben+21], and our method. Our method clearly outperforms any analogies method, at a fraction of the computational cost. The comparisons with [Her+01] were done using an unofficial implementation of their method.

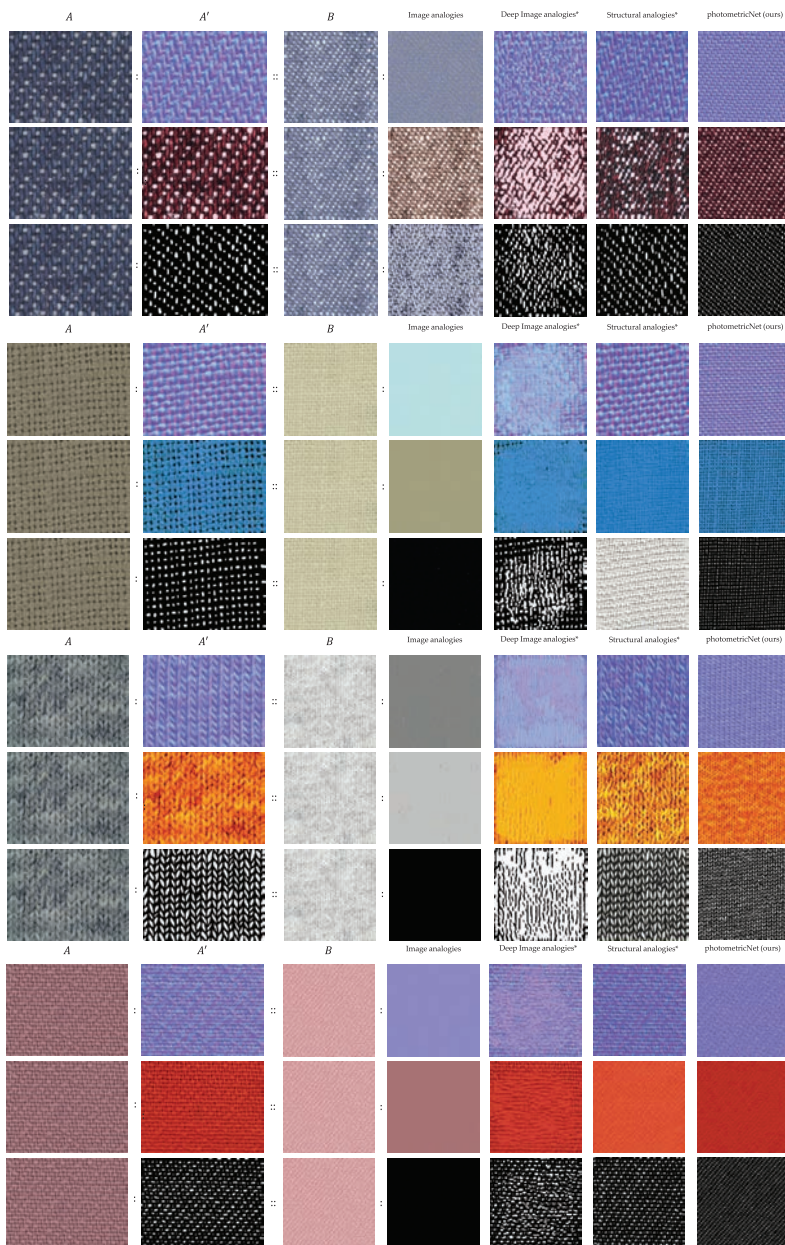


Figure 2.23: Comparison of our method with different configurations of [Tex+20b]. On the top row, we compare their results and ours using a dataset containing only one image, captured with diffuse lighting. On the bottom row, we show results when training these methods with a full photometric dataset. Our method is able to generalize better to different illumination and geometric conditions, with a smaller computational overhead.

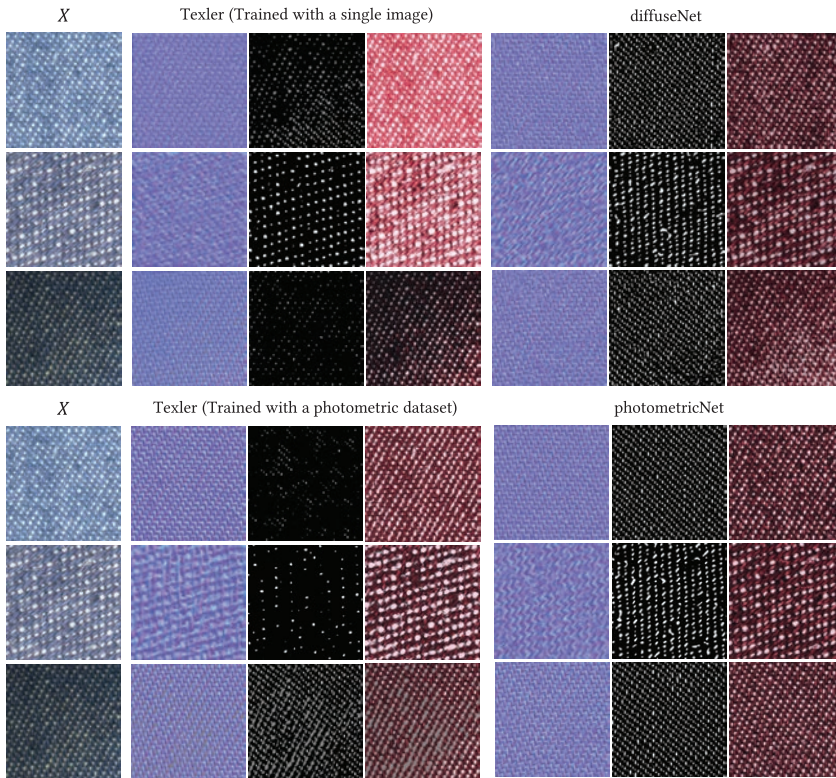
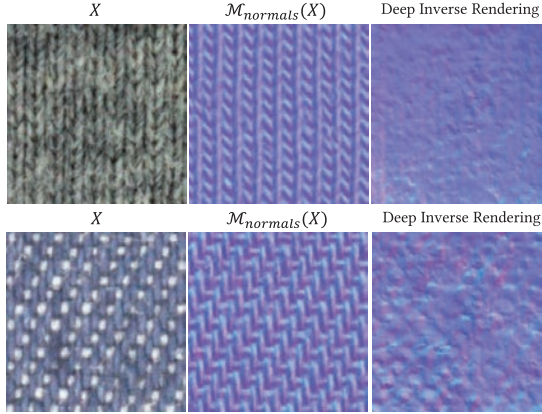


Figure 2.24: Comparison of our method with [Gao+19]. From left to right, we show the input guidance X , the output of photometricNet $\mathcal{M}_\omega$ trained to transfer normal maps, and the output of [Gao+19] after 5000 optimization steps. Their method relies on a pre-trained network, which learns a prior over materials that does not generalize to examples outside their training set. The images in this experiment are of 256×256 pixels, to match the input requirements in [Gao+19].



2.C Extension to SVBSDF Propagation

Figure 2.25: Renders of SVBSDF maps transferred by our extension of photometricNet.



In this section, we present an extension to the previous method for propagating material maps. With it, we can propagate multiple property maps at the same time, learn from many photometric datasets simultaneously, and achieve more accurate

and sharper results. With these improvements, we are able to propagate multiple SVBSDF maps at the same time, allowing for efficient high-resolution material generation, as we show in Figures 2.25 and 2.26. These findings are part of the data generation process which is used to build the datasets we use to train the models described in Section 5 and [Rod+23a], and are also part of a paper on high resolution material digitization, which has been accepted to *SIGGRAPH 2023*:



“Towards Material Digitization with a Dual-scale Optical System” Elena Garces, Victor Arellano, Carlos Rodriguez-Pardo, David Pascual-Hernandez, Sergio Suja, Jorge Lopez-Moreno

ACM Transactions on Graphics (Proceedings of SIGGRAPH) (2023)

2.C.1 Mesoscale Propagation

Figure 2.26: Left, virtual scene rendered using eight different materials digitized with our optical device. Right, real photos of four of the materials taken under diverse illumination conditions: area lights, diffuse lighting, and directional lighting of a high-resolution patch.



To propagate material properties, a common solution is to utilize the estimation of material properties in a localized area and extended it to larger samples [ND06; RSK13; Ste+14; DDB20]. We follow this approach to propagate SVBSDF measurements, and build upon the work presented in Section 2 and [RG21],

which leverage images of the material taken under different illumination conditions to train a neural network that propagates individual maps.

As input to this propagation step, we use a photometric dataset obtained with a microscopic setup, spatially-varying parameters of a micro SVBSDF (Illustrated in Figure 2.28), and a guidance image that will serve as a reference for the mesoscale propagation. We use as guidance the image taken with a mid-range camera and diffuse lighting. Unlike the previous approach, aimed at transferring individual maps, we introduce some modifications to jointly propagate the full stack of parameters. We further adapt the method to be able to take as input more than one micro SVBSDF, necessary if the variability of material is not visible in a single micro capture (see Figure 2.27 (a)). We propose the following modifications:

- 1. Larger Training Dataset.** We improve the generalization of the model by using more training data in two ways: First, we use a larger photometric dataset, using more directional-lit images than what is described in 2.21, as well as diffuse lighting. Second, we use random Gaussian blurs for data augmentation to improve robustness to potential degradation in mesoscale guidance images, and remove affine distortions to the data augmentation policy as they are not required for our SVBSDF transfer problem. Additionally, we train the model for more iterations, with a larger batch size.
- 2. Enable Large Spatially-varying Materials.** We account for materials in which the appearance variability is not observed in a single micro capture by training the propagation network with as many micro SVBSDF as necessary. Figure 2.27 (a) showcases an example that required two microscale captures. Further, we disable the color-invariance data augmentation that randomly shuffles color channels when multiple micro SVBSDFs are available.
- 3. Improved Architecture and Losses.** Finally, we observed that the original architecture was not able to effectively transfer a large number of parameters at the same time. Therefore, inspired by recent work on intrinsic decomposition [Jan+17], texture synthesis [RG22], and material estimation [DLG21], we use a U-Net [RFB15] with a single encoder but a separate decoder for every map we aim to transfer. Further, we introduce residual connections [He+16; Dia+20] into every layer, and substitute Batch Normalization [IS15] for Group Normalization [WH18]. When multiple microscale captures are available, we use a model with additional filters in every layer, to better learn from these extended datasets. This design generates sharper texture maps which preserve the statistics and appearance of each fitted microscale map more accurately. We also added a multi-channel perceptual component [CHB21] to the loss function, which is a powerful

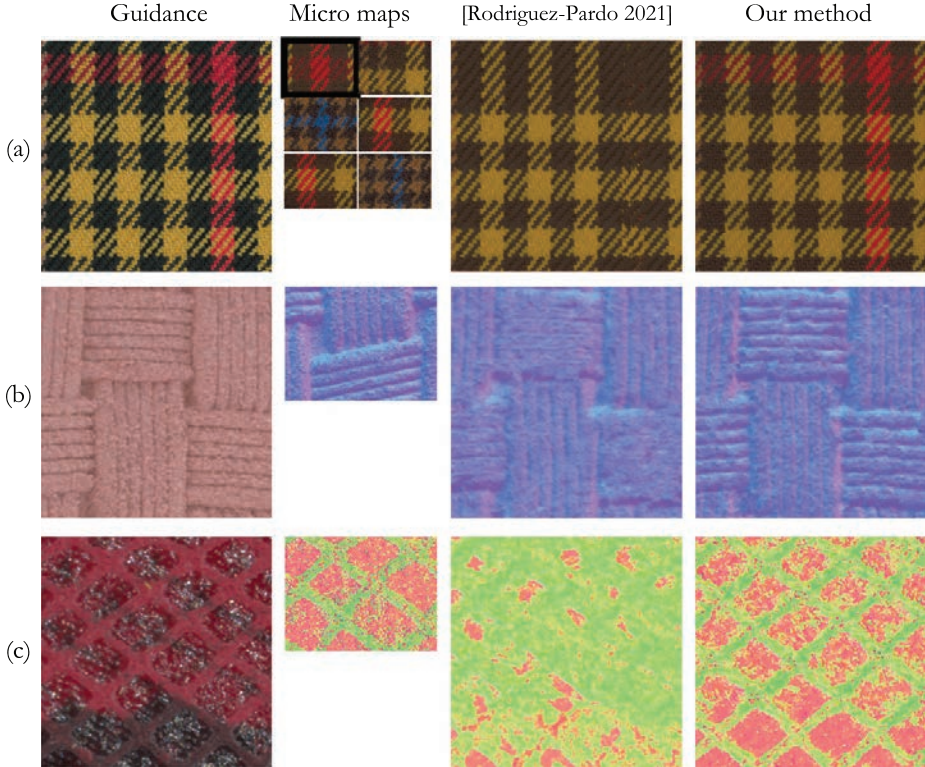
regularizer for texture synthesis. Figure 2.27 (b-c) shows the difference with respect to the previous approach.

2.C.2 Implementation Details

Problem Formulation

As in [RG21], our goal is to transfer spatially-varying attributes of a material captured at microscale to larger samples of the same material. We thus formulate this problem using an image-to-image translation framework. For training, our methods take as input a *set of D photometric datasets* of a material $\mathcal{I}_L^D$, and their pixel-wise corresponding SVBSDFs M_p^D , each comprised of a list of P maps: $p \in \{\text{albedo, roughness, transmittance, IOR, anisotropy, tangents, normals, specularTint, opacity}\}$. Each photometric dataset is comprised of a set of RGB images of the material captured under different illumination conditions: $\mathcal{I}_L^d = \{i_l^d \mid i_l^d \in \mathbb{R}^{n \times m \times 3}\}, |\mathcal{I}_L^d| \geq 1, \forall d \in D$. In contrast to [RG21], which only allowed to learn from a single photometric dataset and to transfer a single attribute, our extended approach allows for learning from multiple datasets $|D| \geq 1$ and multiple property maps $|P| \geq 1$ using a single model. We train a model T which learns to transfer from each image in each photometric dataset to its corresponding SVBSDF: $T(i_l^d) \approx M_p^d, \forall d \in D, l \in L$. During inference, we use a $1 \times p$ guidance image of the material X , which represents a larger sample of it, and estimate its corresponding property maps: $M^X \leftarrow T(X)$. At evaluation time, this guidance image X may be captured with different conditions to those in the training dataset, including different camera or illumination conditions.

Figure 2.27: Comparison with photometricNet [RG21] for mesoscale maps propagation. The first row (a) showcases an example where multiple captures at microscale were needed to cover the spatially-varying albedo of the material. (b) and (c) required a single capture, we show the result of normals and tangents maps compared with previous work.

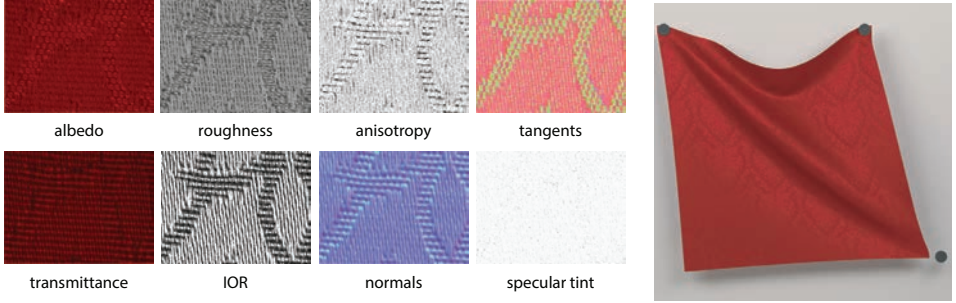


Datasets

We use a more comprehensive photometric dataset to what is proposed in [RG21]. In particular, for a single microscale capture, we use all the light sources used in the fitting algorithm, as well as diffuse-lit microscale images. Additionally, we use both polarization modes separately and, for the same light source, we construct an extra image by averaging the images taken under both polarization configurations. The total size of each photometric dataset is of $|\mathcal{I}_L^d| = 108, \forall d \in D$, four times more data compared with the more limited 27 images proposed in [RG21]. For evaluation, our guidance image is taken with a mid-range camera and *diffuse* illumination, at a resolution of 4072×4072 pixels and a surface area of 10×10 centimeters. We typically use a single photometric dataset ($|D| = 1$). However, for multi-colored or

highly heterogeneous materials in which a single capture is not enough to fully represent the material variability, we capture as many datasets as needed.

Figure 2.28: On the left, microscale SVBSDF maps. On the right, a render generated with the maps propagated with our method.



Loss Function

Our loss function is the combination of two terms: weighted pixel-wise losses for each target map, and a multi-map style loss:

$$\mathcal{L} = \sum_p \lambda_p \mathcal{L}_{pixel_p} + \lambda_{style} \mathcal{L}_{style} \quad (2.3)$$

$\mathcal{L}_{pixel}$ is the L_1 norm weighted per map in the SVBSDF, $\lambda_p \cdot L_1$ produces sharper results than higher-order alternatives [RG21], such as L_2 . For the $\mathcal{L}_{style}$ loss term, we follow recent work on multi-channel texture synthesis, which extends deep perceptual losses for style transfer to the SVBRDF synthesis problem [CHB21]. This component of the loss function acts as a regularizer and allows the model to generate maps that better preserve the appearance of the ground truth training data.

Training Implementation Details

Model Design We use a lightweight U-Net [RFB15] architecture, with a few modifications based on recent work to maximize its efficiency and the quality of its outputs. We specify the full model architecture and model sizes in Figure 2.29. In every convolutional block of the model, we use residual connections [He+16; Dia+20], for better training convergence and preservation of details in the input images. We use 1×1 convolutions on these residual connections. We use a single decoder for each map, so as to maximally preserve their individual appearance and statistics. This has been proposed in recent work on texture synthesis [RG22], material capture [DLG21] or intrinsic decomposition [Jan+17]. On the convolutional blocks, we leverage Group Normalization [WH18]. Upsampling is done using transposed convolutions. As shown in Figure 2.29, the model has four hidden layers on the encoder and each decoder, however, we vary the size of those layers

depending on whether the model is trained with a single microscale capture (for which we use a width factor of $W = 16$) or multiple ($W = 32$). Every other implementation detail in the model (stride, bias, pooling) follows [RFB15; RG21].

Training and Evaluation We use PyTorch [Pas+19], Torchvision [MR10], and Kornia [Rib+20] for training. We leverage mixed precision training and automatic gradient scaling [Mic+18], to accelerate the training process and regularize the models. Optimization is done using Adam [KB15] for 5000 iterations, with a learning rate of 0.002, a batch size of 40, and no weight decay. This process takes around 15 minutes when training with a single microscale capture, and 25 when using multiple captures as input. We evaluate the guidances images using half precision. We measure these times on an NVIDIA 1080Ti GPU.

Data Augmentation We follow the same training procedure defined in [RG21]. However, we do not use random rotations or shears, and remove the color invariance data augmentation whenever multiple microscale captures are available. Further, we introduce random Gaussian Blurs for data augmentation, using a $p = 0.5$, a kernel size of 5, and sigma selected uniformly at random for each element in each batch: $\sigma \sim U(0.1, 11)$. As in [RG21], we also perform random cropping, using crops of 128×128 pixels. For each element in each batch during training, we randomly choose the photometric dataset $d \in D$, and, from it, a random light source $l \in L$.

Loss Function We weight the loss function as follows: For the pixel wise loss: $\lambda_{\text{normals}} = 3, \lambda_{\text{IOR}} = 1, \lambda_{\text{roughness}} = 1, \lambda_{\text{albedo}} = 3, \lambda_{\text{tangent}} = 3, \lambda_{\text{anisotropy}} = 1, \lambda_{\text{specular}} = 1, \lambda_{\text{Tint}} = 1, \lambda_{\text{transmittance}} = 3, \lambda_{\text{opacity}} = 1$. The perceptual component [CHB21] is weighted with $\lambda_{\text{style}} = 0.25$, and we use the *AlexNet* variant of *LPIPS* [Zha+18] as our backbone, which provides a powerful yet lightweight loss for texture transfer, as shown in [RG21].

2.C.3 Results

In Figure 2.30, we show an ablation study on the improvements of the maps propagation method proposed in [RG21]. We build upon their proposed implementation, removing their random shifts and rotations, and make progressive changes, in order, to the dataset size, model architecture, loss function, and data augmentation. We observe high-quality propagations, but the baseline sometimes produces overly smooth outputs (Fleece, Leather-Brown, Rib-Silver, or Leather-Lizard). Our proposed increased dataset and model architecture modifications enhance the maps' sharpness, and our perceptual loss function allows for the propagation of additional details. Our final model removes the color augmentation proposed in [RG21] whenever multiple microscale captures are available, and introduces random blurs during training, allowing for achieving the highest quality results and for accurate base color propagations in challenging cases, as in Tartan.

Figure 2.29: A full diagram of the model architecture we use for our mesoscale propagation problem. Building upon [RG21], use a lightweight U-Net [RFB15] architecture. Following previous work [Jan+17; RG22; DLG21], we use a different decoder for every map we aim to transfer. Each of our convolutional blocks, illustrated on the right, contain residual connections [He+16; Dia+20] and Group Normalization [WH18]. In red, we show the input/output dimensions (spatial, channels) of each layer; in blue, convolutional blocks and layers; in green, upsampling and concatenating operations; in yellow, normalization layers; and in purple, regularizations, and non-linearities. We set W to $W = 16$ when a single microscale capture is available as training data, $W = 32$ otherwise.

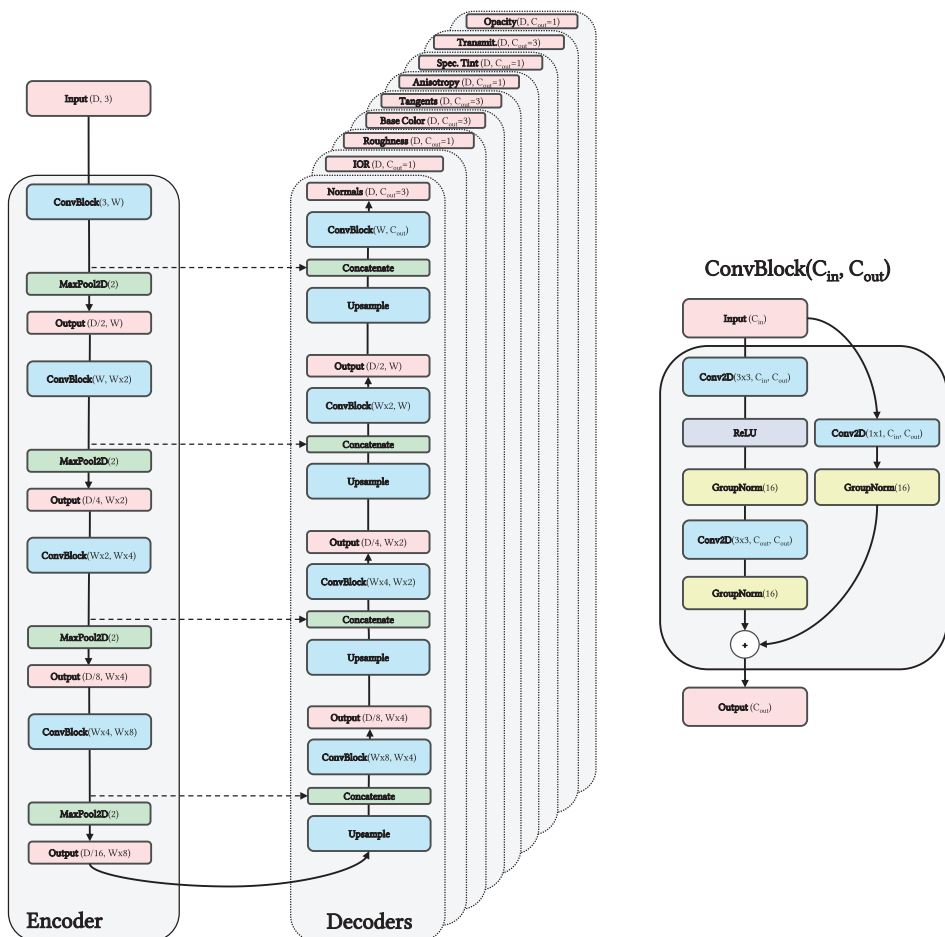
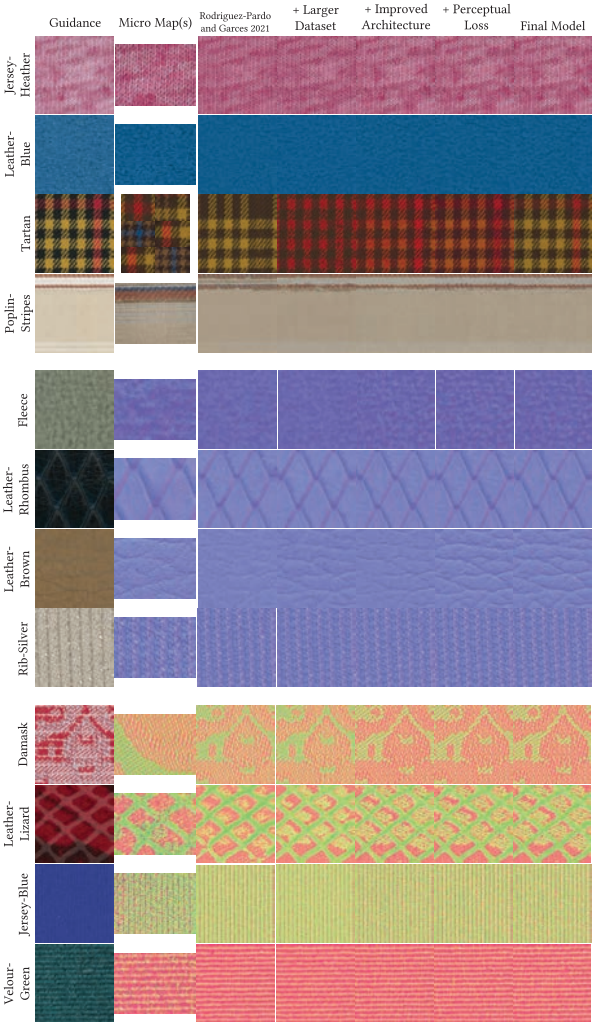


Figure 2.30: Ablation study of our proposed improvements with respect to tphotometricNet [RG21], on base color (top rows), normals (mid) and tangent map (bottom) transfer. We start from the implementation in [RG21], which shows adequate but smooth results. On its right, we show the impact of increasing the dataset size, which tends to generate higher-quality estimations (Leather-Brown).

With our improved architecture (fifth column), we achieve sharper maps (see Leather-Lizard, Rib-Silver, or Fleece). Using a perceptual loss (sixth column), we further improve the maps quality (Leather-Brown). Our final model (last column), which removes the color augmentation in [RG21] and introduces random blurs to the model training, achieves the highest quality, and allows for accurate base color transfers, as shown in Tartan. Input guidance images and training micro maps are shown on the first and second columns, respectively.



CHAPTER 3. A SINGLE IMAGE GENERATIVE MODEL FOR TILEABLE TEXTURE SYNTHESIS

Figure 3.1: Results obtained with SeamlessGAN for tileable texture synthesis. We show 2×2 repetitions of the outputs on the squared images, and the input exemplars by their side.



In this chapter, we present *SeamlessGAN*, a method capable of automatically generating tileable texture maps from a single input exemplar. In contrast to most existing methods, focused solely on solving the synthesis problem, our work tackles both problems, *synthesis* and *tileability*, simultaneously. Our key idea is to realize that tiling a latent space within a generative network trained using adversarial expansion techniques produces outputs with continuity at the seam intersection that can then be turned into tileable images by cropping the central area. Since not every value of the latent space is valid to produce high-quality outputs, we leverage the discriminator as a perceptual error metric capable of identifying artifact-free textures during a sampling process. Further, in contrast to previous work on deep texture synthesis, our model is designed and optimized to work with multi-layered texture representations, enabling textures composed of multiple maps such as albedo, normals, etc. We extensively test our design choices for the network architecture, loss function, and sampling parameters. We show qualitatively and quantitatively that our approach outperforms previous methods and works for textures of different types. The contributions presented in this chapter have led to the following publication¹:

¹ Additional publication details and results are included on [the project website](#).



"SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps" Carlos Rodriguez-Pardo, Elena Garces

IEEE Transactions on Visualization and Computer Graphics (TVCG)

(2022)

3.1 Introduction

Realistic and high-quality textures are important elements to convey realism in virtual environments. These can be procedurally generated [Tu+20; HDR19; Gue+20; Gal+12; Gil+14; Gui+17; HN18], captured [Guo+20b; LSC18] or synthesized from real images [EF01; Kwa+05; Zho+18; Mor+17]. Frequently, textures are used to efficiently reproduce elements with repetitive patterns (for example, facades, surfaces, or materials) by means of spatially concatenating -or *tiling*- multiple copies of themselves. Creating tileable textures is a very challenging problem, as it requires a semantic understanding of the repetitive elements, often at multiple scales. For this reason, such a process is frequently done manually by artists in 3D digitization pipelines.

Recent advances in Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) have been applied to texture synthesis problems [Zho+18; FAW19; Liu+20; BJV17; JBV16; Mar+20; Her+20] showing unprecedented levels of realism and quality, however, the output of these methods is not tileable. Despite recent methods [DH19; BJV17; Rod+19; Mor+17; Li+20; Nik+21] addressing the problem of tileable texture synthesis, we show that they either assume a particular level of regularity or the generated textures lose a significant amount of visual fidelity with respect to the input exemplars. Further, most of these methods have only focused on synthesizing single images. Rendering realistic materials requires more information about their optical properties beyond what represents a single RGB pixel. To this end, it is common to use spatially-varying BRDFs [Des+19], which are optical appearance models parameterized by stacks of images, each one representing a different property, such as albedo, normals, or transparency. As the number of methods that generate texture stacks from physical samples grows [Des+19; DDB20; Guo+20b] so does the need to turn them into *tileable texture stacks*.

In this chapter, we propose a deep generative model, *SeamlessGAN*, capable of synthesizing tileable texture stacks from inputs of arbitrary content. In contrast

to *Wang Tiles* [Coh+03], by which a single large texture region is created by concatenating multiple different tiles with matching borders, we aim to automatically obtain a seamless single-tile, which allows for reduced memory consumption and enhanced usability in casual scenarios when the user might lack the necessary artistic skills. Our key idea is to realize that tiling a latent space within a generative network produces outputs with continuity at the seam intersection [FAW19], which can then be turned into tileable images by cropping the central area. Since not every value of the latent space is valid to produce high-quality outputs, we follow a double strategy: First, we train the generative network using an adversarial expansion technique [Zho+18], which provides a latent encoding of the input texture, which can then be decoded into high-quality outputs that double the spatial extent of the input. Second, we use the trained discriminator as a local quality metric in a sampling algorithm. This allows us to find the input of the generative process that produces tileable textures similar to the original, as well as multiple candidates per input exemplar. As opposed to previous work, which focused on maximizing stationarity [Mor+17] and thus might remove important high-level texture features in regular or near-regular textures, our method is focused on maximizing tileability while preserving the original texture as intact as possible, in terms of its stylistic and semantic properties. To allow for the synthesis of stacks of textures, we propose a neural architecture composed of various decoder networks. Despite not explicitly imposing inter-map consistency, we show that it is implicitly guaranteed by how the generative network is trained. Without losing generality, we show texture stacks synthesis of two maps: an albedo and a normal map. We demonstrate that our method outperforms state-of-the-art solutions on tileable texture synthesis of single images and show several examples for synthesizing tileable texture stacks. We validate our design choices through several ablations studies and off-the-shelf perceptual quality metrics.

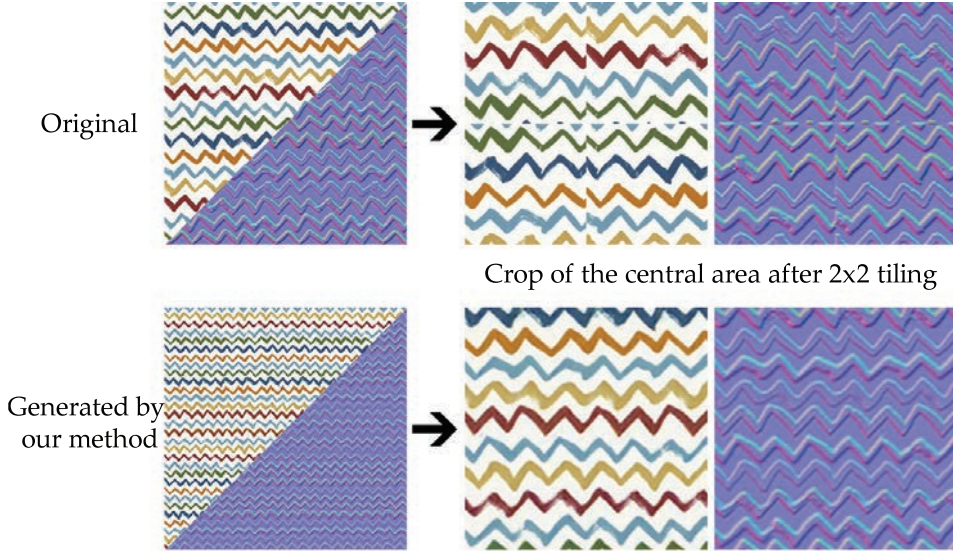
3.2 Related work

We review the texture synthesis methods most closely related to our work. For a more comprehensive review, please refer to the surveys in [Akl+18; Raa+18]. We will also mention other related work regarding tileable texture synthesis, and deep internal learning.

3.2.1 Texture Synthesis

Traditionally, **non-parametric texture synthesis** algorithms worked by ensuring that every patch in the output textures approximates a patch in the input texture. Earlier methods included image quilting [EF01; EL99], GraphCuts [Kwa+03], genetic algorithms [DZP07], and optimization [Kwa+05; PS00]. More recent approaches use variations of PatchMatch [Bar+09; Bar+15] as a way of

Figure 3.2: Naïvely tiling the original texture causes discontinuities at the seam intersections as shown in the top row. Our method automatically generates a tileable texture stack from an input exemplar which double the size of the input.



finding correspondences between generated and input images [Kas+15; Dar+12; Zho+17].

Despite those methods showing high-quality results for textures of different characteristics, recent work on deep parametric texture synthesis shows better generality and scalability, requiring less manual input. Our approach belongs to the category of **Parametric texture synthesis**. These methods learn statistics from the example textures, which can then be used for the generation of new images that match those statistics. While traditional methods used hand-crafted features [De 97; HB95], recent parametric methods rely on deep neural networks as their parameterization. Activations within latent spaces in pre-trained CNNs have been shown to capture relevant statistics of the style and texture of images [GEB16a; Gat+17; JAF16; Ren+17; RB19]. Textures can be synthesized through this approach by gradient-descent optimization [Sne17; GEB15a] or by training a neural network that learns those features [DB16; Nik+21]. Finding generic patterns that precisely describe the example textures is one of the main challenges in parametric texture synthesis. Features that describe textured images in a generic way are hard to find and they typically require hand-tuning. Generative Adversarial Networks (GANs), which have shown remarkable capabilities in image generation in multiple domains [Kar+18; KLA19; Kar+20b], can learn those features from data. Specifically, in texture synthesis, they have proven successful at generating new samples

of textures from a single input image [Zho+18] or from a dataset of images [BJV17; FAW19; JBV16; Liu+20; Mar+20]. We build upon the method of Zhou *et al.* [Zho+18] which shows good performance on the synthesis of non-stationary single image textures, and extend it to synthesize texture stacks, as well as generate tileable outputs.

3.2.2 Texture Tileability

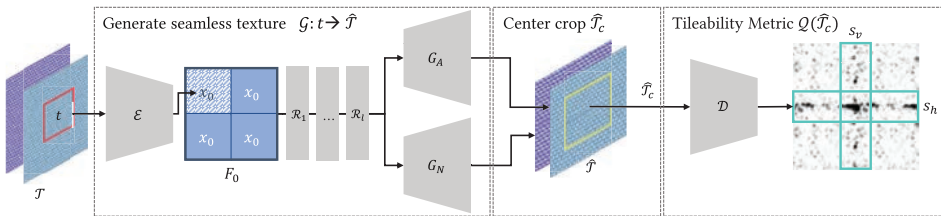
Synthesizing tileable textures received surprisingly little attention in the literature until recent years. Moritz *et al.* [Mor+17] propose a non-parametric approach that is able to synthesize textures from a single example while preserving its *stationarity*, which measures how tileable the texture is. Li *et al.* [Li+20] propose a GraphCuts-based algorithm. They first find a patch that optimally represents the texture, then use graph cuts to transform its borders to improve its tileability. This method allows for synthesizing multiple maps at the same time. Relatedly, Deliot and Heitz [DH19] propose a histogram-preserving blending operation for patch-based synthesis of tileable textures, particularly suited for stochastic textures. The power of deep neural networks for tileable texture synthesis has also been leveraged in the past years. First, Rodriguez-Pardo *et al.* [Rod+19] exploit latent spaces in a pre-trained neural network to find the size of the repeating pattern in the input texture. Then, they use perceptual losses for finding the optimal crop of the image such that, when tiled, looks the most similar to the original image. Also leveraging deep perceptual losses and using Neural Cellular Automata [Mor+20] as an image parameterization, Niklasson *et al.* [Nik+21] generate self-organizing textures that are seamlessly tileable by design, but are limited in resolution and by the quality of the gram matrix perceptual metric used as a loss function [Hei+21]. Bergmann *et al.* [BJV17] achieve tileability in their output textures by training a multi-image GAN in a periodic spatial manifold. Our proposed method does not follow any of these approaches. Instead, we build upon a state-of-the-art single-image texture synthesis method, which is able to generate high-quality and high-resolution images, and extend it to generate textures which are tileable. To this end, we propose a sampling algorithm that finds the input of the generative process that maximizes a novel tileability metric. Our goal is to preserve the input original texture as intact as possible while imposing artifact-free continuity at the intersection of the seams when the texture is tiled. Furthermore, our method has a reduced computational footprint compared to other deep generative texture synthesis methods, as we show in the results section.

3.2.3 Deep Internal Learning

Learning patterns from a single image has been studied in recent years, in contexts different to those of texture synthesis. Ulyanov *et al.* [UVL18] show

that single images can be represented by randomly initialized CNNs, and show applicability on denoising or image inpainting problems. A similar method is proposed by Shocher *et al.* [SC18] for single-image super-resolution. Additionally, single images have shown to be enough for learning low-level features that generalize to multiple problems [ARV19]. Hypernetworks [HDL16] in *Implicit Neural Representations* [Sit+20; Tan+20] also allow for shift-invariant priors over images, which can then be used in generative models [DTD21]. Single-image generative models have been explored for domains other than textures. GANs trained on a single image have been used for image retargeting [Sho+19], deep image analogies [Ben+21], or for learning a single-sample image generative model [SDM19; Hin+21; Sho+19; SGK21; VZH20]. These methods, while powerful for natural images, are not well-behaved for textures, as shown in [Liu+20] and [Mar+20]. By introducing inductive biases specially designed for textured images, characterized by their repeating patterns, deep texture synthesis methods achieve better performance in textured images than generic single-image synthesis approaches, which need to account for more globally coherent semantic patterns.

Figure 3.3: Overview of *SeamlessGAN*. A crop t of the input texture stack T , is fed to an encoder E , which transforms it into a latent space $x_0 \leftarrow E(t)$. We tile this latent space vertically and horizontally, obtaining a latent field F_0 . F_0 is further processed by several residual blocks $R_i, i \in \{1, l\}$. The resulting latent variables are transformed by two different decoders, G_A and G_N , which output 4 copies of a candidate tileable texture $\hat{T}$. By cropping the center part of this texture, we obtain a single of those copies, with seamless borders, $\hat{T}_c$. Additionally, this texture can be analyzed by a discriminator D which provides local estimations of the quality of the synthesis. We introduce a tileability evaluation function Q , which, by analyzing two vertical and horizontal search areas s_v, s_h , is able to detect artifacts that may arise when tiling the texture. This gives us an estimation of how *tileable* the texture is. This estimation can then be used by a sampling algorithm for generating high-quality tileable textures.



3.3 Overview

Our method takes as input an *untileable texture stack* T , which is a layered representation—or SVBRDF [BS12]—of a material used by render engines to virtually reproduce it. A typical texture stack is composed of several maps such as albedo, normals, or roughness. For simplicity, and without losing

generality, we assume that our texture stacks have two maps: an albedo A , and a normal map N , both are RGB images of the same dimensions. There are several methods that generate texture stacks for any material using, for example, smartphones [Des+19; DDB20; Guo+20b; Shi+20; VPS21; Rod+23a], however, these stacks are not tileable by default. Having this data as input, we propose a two-step automatic pipeline to generate tileable texture stacks from arbitrary material input exemplars.

In the first step, described in Section 3.4, we use crops t of the input texture T to train a Generative Adversarial Network (GAN) able to synthesize novel *untileable* texture stacks $\hat{T}$ using adversarial expansion [Zho+18]. This training framework has been shown to provide state-of-the-art results on single-sample texture synthesis, surpassing previous approaches [Uly+16; BJV17; Hen+21]. Thanks to using a GAN, we learn an implicit representation of the texture parameterized in two neural modules: G and D . $G: t \rightarrow \hat{T}$ is a generator that outputs new untileable textures which double the spatial resolution of the input. D is a discriminator that, thanks to the adversarial framework used for training, is able to distinguish real from fake textures.

In the second step, described in Section 3.5, we produce tileable stacks by means of two key ideas: first, we tile a latent space x_0 of the trained generator G , obtaining a novel texture stack showing continuity at the seams intersection $\hat{T}$. Second, we implement a sampling process using the trained discriminator D used as quality metric Q to find an optimally tileable texture stack $\hat{T}^*$. An overview of our sampling step is shown in Figure 3.3.

In the following, we first describe our GAN architecture, including our proposal to deal with textures stacks. Then, we describe the sampling process to generate tileable ones.

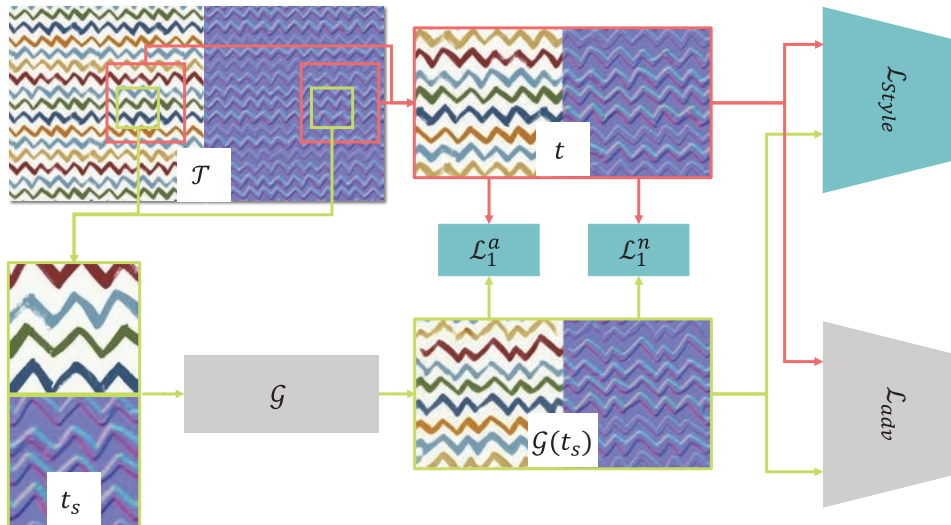
3.4 Self-Supervised Texture Synthesis with Adversarial Expansion

For each input texture T we train a GAN, whose generator G is able to synthesize novel examples $\hat{T}$. Unlike other GAN frameworks, which take as input a random vector, we use crops t of the original stack as input to guide the generative sampling, such that, $\hat{T}^t = G(t)$ for $t \in T$. The training strategy builds upon the work of Zhou *et al.* [Zho+18], which uses *adversarial expansion* to train the network as follows: First, a target crop $t \in T$ of $2k \times 2k$ pixels is selected from the input stack. Then, from that target crop t , a source random crop $t_s \in t$ is chosen with a resolution of $k \times k$ pixels. The goal of the generative network will be to synthesize t given t_s . This learning approach is fully self-supervised. The generative model is trained alongside a discriminator D , which learns to predict whether its inputs are the target crops $t \in T$ or the generated samples $\hat{T} = G(t_s)$. Figure 3.4 shows an overview of the training strategy.

3.4.1 Network Architecture

SeamlessGAN is comprised of an encoder-decoder convolutional generator G with residual connections, and a convolutional discriminator D . So as to be able to synthesize textures of multiple different sizes, the networks are designed to be fully-convolutional. We follow the residual architectural design in [Zho+18], with two extensions: First, building on recent advances in style transfer algorithms, we use *Instance Normalization* [UVL16] before each ReLU non-linearity in the generator and each Leaky-ReLU [MHN13] operation in the discriminator. This allows to use normalization for training the networks without the typical artifacts caused by Batch Normalization [Cho+18; Des+18; NK18; Kur+19]. Second, in order to allow for the synthesis of multiple texture maps at the same time, we propose a variation in the generator architecture. In the following, we describe the details of each component, with a particular focus on the elements that are different from [Zho+18].

Figure 3.4: An overview of our training framework for learning to synthesize texture stacks through adversarial expansion. At each iteration, from an input stack T , we randomly crop a target crop t , from which we select a source crop t_s . The goal of the generator G is to estimate t given t_s . We measure the difference between target and estimated crops using a combination of adversarial, pixel-wise, and perceptual loss functions.



Generator

The generative architecture proposed is comprised of three main components: An *encoder* E , which compresses the information of the input texture t into a latent space, $x_0 \leftarrow E(t)$, with half the spatial resolution of the input texture. Then,

a set of *residual blocks*, $R_i, i \in \{1, l\}$, which learn a compact representation of the input texture $x_i \leftarrow R_i(x_i - 1) + x_i - 1$. Finally, a stack of *decoders* $\mathbf{G}$ that transforms the output of the last residual block RI into an output texture $\hat{T} \leftarrow \mathbf{G}(x_l)$. Residual learning allows for training deeper models with higher levels of visual abstraction and which generate sharper images [He+16; RFB15; Iso+17]. Similar residual generators have been used in unsupervised image-to-image translation problems [Zhu+17a].

Synthesizing a texture stack of multiple maps with a single generative model poses extra challenges over the single map case. Each map represents different properties of the surface, such as geometry or color, resulting in visually and statistically different images. This suggests that independent generative models per map could be needed. However, the texture maps must share pixel-wise coherence, which is not achievable if multiple generative models are used. Inspired by previous work on intrinsic images [Jan+17; YS19; LS18], we propose the use of a generative model that learns a shared representation of all the texture maps, but has different decoders for each of them. The latent space within the generator will thus encode a high-level representation of the texture, which is then decoded in different ways for each different map in the texture stack. Specifically, we train a stack of decoders, $\mathbf{G} = \{G_A, G_N\}$, one for each map of the stack (albedo and normals in our case).

Discriminator

For the discriminator D , we use a *PatchGAN* architecture [Zhu+17a; Iso+17; Led+17; Zho+18], which, instead of providing a single estimation of the probability of the whole image being real, it classifies this probability for small patches of it. This architecture has several advantages for our problem. First, it provides local assessments of the quality of the synthesized textures, which we exploit for obtaining high-quality textures. Second, its architecture design allows to provide some control over what kind of features are learned: by adding more layers to D , the generated textures are typically of a higher semantic quality but local details may be lost. A comprehensive study on the impact of the depth of D can be found in [Zho+18].

Loss Function

We train the networks following a standard GAN framework [Goo+14b]. We iterate between training D with a real sample texture stack T and a generated sample $\hat{T}$. The adversarial loss L_{adv} is extended with three extra loss terms: $\mathcal{L}_1^a$, $\mathcal{L}_1^n$, and L_{style} ; corresponding respectively to the pixel-wise distance between the generated and target albedo, normals, and a perceptual loss. The perceptual loss L_{style} is computed as the sum of the perceptual losses between target and generated albedo maps, and target and generated normal maps. We follow the *Gram Loss* described in [GEB16b] as our perceptual loss. We weight the total

style loss by weighting different layers in the same way described in [Rod+19; Zho+18; GEB16b]. Our global loss function thus is defined as:

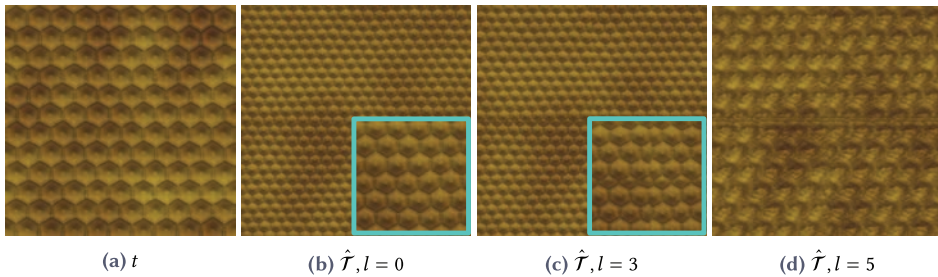
$$\mathcal{L} = \lambda_{adv} \mathcal{L}_{adv} + \lambda_{\mathcal{L}_1^a} \mathcal{L}_1^a + \lambda_{\mathcal{L}_1^n} \mathcal{L}_1^n + \lambda_{style} \mathcal{L}_{style} \quad (3.1)$$

3.5 Tileable Texture Stack Sampling

3.5.1 Latent Space Tiling

After training, the generator G is able to synthesize novel samples of the texture given a small exemplar of it. Although these novel samples duplicate the size of the input, they are not tileable by default. Previous work [FAW19] showed that by spatially concatenating different latent spaces in a ProGAN [Kar+18] generator, it is possible to generate textures that contain the visual information of those tiles while seamlessly transitioning between the generated tiles. Inspired by this idea, we spatially repeat (horizontally and vertically) the first latent space $\mathbf{x}_0$ within the generative model, obtaining a latent field $\mathbf{F}_0$. This field is passed through the residual layers R_l and the decoders $\mathbf{G}$ to get a texture stack, $\hat{T}$, that contains four copies of the same texture with seamless transitions between them (see Figure 3.3). At first, the latent field $\mathbf{F}_0$ shows strong discontinuities at the borders between the four copies. Later, as the texture is passed through the network, these artifacts are progressively transformed into seamless borders by the rest of the residual blocks and the decoder of the model. The intuition behind this idea is that, after training the network with the target texture, each of these latent spaces encode low resolution versions of the input in the spatial domain and semantically-rich information in the deeper layers.

Figure 3.5: Impact of the latent field level l on the quality of the output textures $\hat{T} = G(t)$, using the same input t . Generating the latent field $\mathbf{F}_l$ early layers ($l = 0$) generates the best visual results, later layers either create small artifacts ($l = 3$) or generate unrealistic textures ($l = 5$). A zoom of the central crop is shown as an inset.

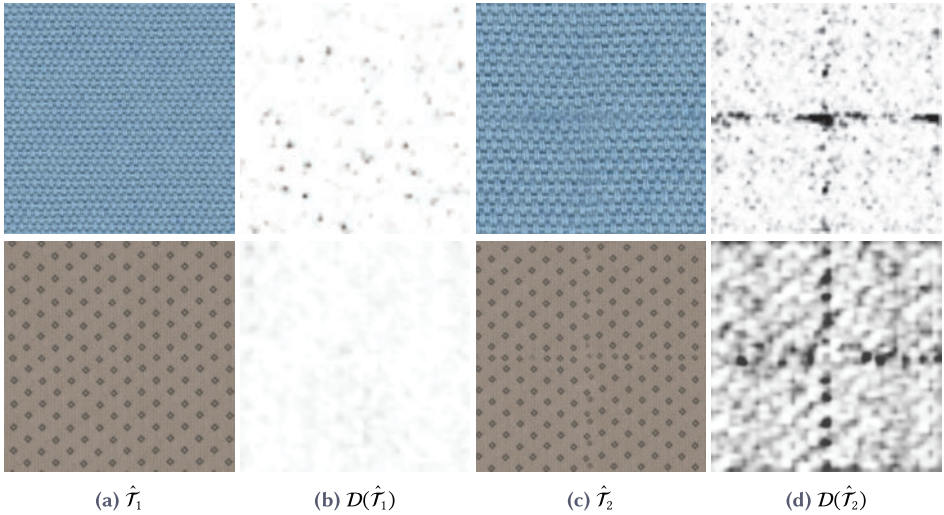


A seamlessly tileable texture stack $\hat{T}_c$ is obtained by cropping the central region (with an area of 50% of $\hat{T}$). The predicted texture $\hat{T}$ has 4×4 the

resolution of the input crop t . Thus, after the cropping operation, $\hat{T}_c$ has twice the resolution of the input t .

A key parameter to select is the level l at which to split the generative process. As shown in Figure 3.5, generating the latent field F by tiling earlier levels of the latent space forces the network to transform F more times, thus resulting in more seamlessly tileable textures. We confirm the results found in [FAW19] and noticed that generating this latent field at earlier levels of the latent space ($l \in \{0, 1\}$) yields the best visual results, by allowing for smoother and more semantically coherent transitions between tiles. We thus tile the output of the encoder $x_0 = E(t)$, before transforming it by the residual blocks R_l .

Figure 3.6: Outputs of D for textures of different qualities. (a) The generated albedos $\hat{T}_1$ are artifact-free on the search areas (b), thus the discriminator $D(\hat{T}_1)$ is fooled to believe $\hat{T}_2$ cannot fool the discriminator, which finds artifacts on the central areas $D(\hat{T}_2)$ (d).



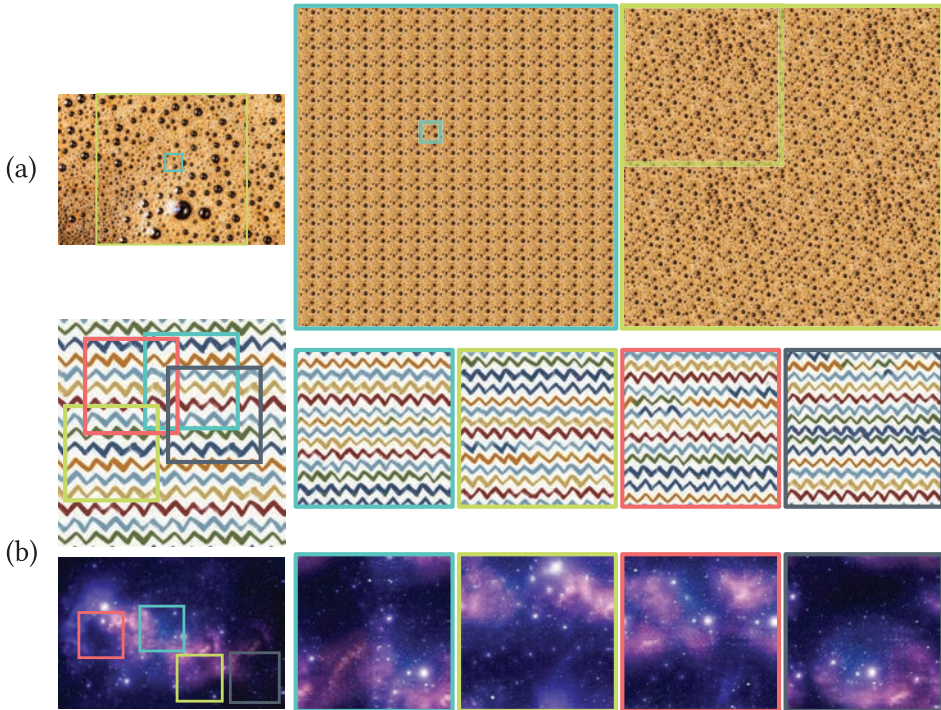
3.5.2 Discriminator-guided Sampling

Our strategy to tile the latent space guarantees that the generated texture is a continuous function with smooth transitions between the boundaries of the tiles. However, in contrast to the algorithm in [FAW19], where the latent spaces are drawn from random vectors, ours are encoded representations of input textures. Thus, the selection of the input that the network receives plays an important role on the quality of the output textures. As shown in Figure 3.6, not all the generated textures are equally valid. This selection can be posed as an optimization problem: $t^* = \operatorname{argmax}_t Q(G(t))$, where the goal is to find the crop

t^* that maximizes the quality Q of the generated texture $G(t)=\hat{T}_c$. To solve this optimization problem, one option could be to pose it as an optimization in the generative latent space. However, we found that a simpler solution based on sampling already provides satisfactory results. The remainder of this section explains the sampling process and the quality metric.

Sampling By using a fully-convolutional GAN, our model can generate textures of any size, hardware being the only limiting factor. This is key for tileable texture synthesis as, even if a given texture is seamlessly tileable, larger textures require fewer repetitions to cover the same spatial area, which ultimately results in fewer repeating artifacts, as illustrated in Figure 3.7 (a). There are two main challenges when finding tileable textures: the input texture needs to contain the distribution of the existing repeating patterns, and the input tile itself must not create strong artifacts when tiling the latent spaces of the generator.

Figure 3.7: *SeamlessGAN* can generate multiple tileable outputs from the same sample T . By cropping different parts of T , we can feed the generator G with different inputs t_c (shown on the left), generating tileable textures $\hat{T}_c$ of different sizes. On the example at the top (a), the synthesized images are tiled so they cover the same spatial extent, which shows that, even if the textures are tileable, larger textures generate fewer repeating artifacts. (b) Shows the tile resulting from sampling different parts of the input image.



Our goal is thus to find the largest possible tileable texture stack. To do so, we sample multiple candidate crops for a given crop size $c \in \{c_{\min}, c_{\max}\}$ is the resolution of the input T . We sample crop sizes starting at the largest possible size $c = c_{\max}$ and stop when we find a suitable candidate according to the tileability metric Q . As shown in Figure 3.7 (b), this sampling mechanism also allows us to generate multiple tileable candidates for a single exemplar if we choose different parts of T as input.

Discriminator-guided Quality Function, Q The second component of our sampling strategy is the quality function used to determine whether a stack is tileable or not. We observed that the artifacts appear on vertical and horizontal frames around the center of the textures (Figure 3.6). This is likely caused by strong discontinuities or gradients on the same areas of the tiled latent spaces, which the rest of the generative network fails to transform into realistic textures. Following recent work on generative models [SSK20], we use the discriminator D as a semantic-aware error metric that can be exploited for detecting local artifacts in the generated textures. This can be done in our case because the global loss function contains pixel-wise, style, and adversarial losses. The adversarial loss learns the semantics of the texture, whereas the L_i and style losses model color distances or repeated patterns [Rod+19]. This combination of loss functions allows us to balance textural, semantic, and perceptual color properties.

We thus design a quality evaluation function Q that estimates if the generated texture stack $\hat{T}_c$ is *tileable*, looking for artifacts on a central area, $S \in D(\hat{T}_c)$ of the discriminator. This area is composed of two regions, $S = S_v \cup S_h$, where S_v is a vertical area, and S_h is an horizontal one, both centered on the output of the discriminator (Figure 3.3). The function Q leverages the fact that D outputs 0 when it believes a patch to be synthetic. However, as the values are sample-dependent, we establish a threshold $\tau \in \mathbb{R}$ using the values of the rest of the image S as a reference $\tau = \gamma \cdot \min(S_r)$, where $S_r = D(\hat{T}_c) \setminus S$ is the remaining part of the image, and $\gamma \in \mathbb{R}$ is a threshold that allows to control the sensitiveness of Q . Consequently, $Q(\hat{T}_c)$ is 1 if $\min(s_v)$ and $\min(s_h)$ are greater or equal than τ , considering the texture as *tileable*, and 0 otherwise. The goal of Q is to estimate whether or not the central areas, where discontinuities may be present, are as realistic as the rest of the texture. It works as a classification function which returns a high value if the texture is identified as real by the discriminator in such areas. This allows us to create a sampling strategy that distinguishes between images with local artifacts from seamless textures. By using minimum values instead of the mean estimation of the discriminator, our quality metric focuses on detecting artifacts which may arise during the latent space tiling operation.

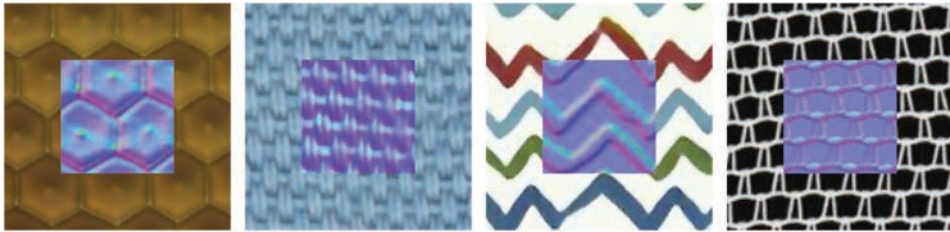
3.6 Implementation Details

As described in Section 3.4, we follow a standard GAN training framework, by iterating between training the discriminator and the generator. We use a batch

size of 1, and an input size of $k = 128$. All weights are initialized by sampling a Gaussian distribution $N(0, 0.02)$, following standard practice [Zhu+17a]. G has $l = 5$ residual blocks with ReLU activations, and D is comprised of 5 convolutional layers with Leaky-ReLU non-linearities, and a Sigmoid operation at the end. We use a stride of 2 for the downsampling operations in E and transposed convolutions [LSD15] for upsampling in the decoders G . We weight each part of the loss function as: $\lambda_{\text{adv}} = \lambda_{\text{style}} = 1$, and $\lambda_{\mathcal{L}_1^g} = \lambda_{\mathcal{L}_1^d} = 10$.

The networks are trained for 50000 iterations using Adam [KB15], with an initial learning rate of 0.0002, which is divided by 5, after iterations 30000 and 40000. Aside from random cropping, we do not use any other type of data augmentation method. The models are trained and evaluated using a single NVIDIA GeForce GTX 1080Ti. Even if training takes around 40 minutes for each texture stack, once trained, the generator can generate individual new samples in milliseconds, before checking for tileability. We use PyTorch [Pas+19] as our learning framework. To accelerate the training process, we leverage mixed precision training and automatic gradient scaling [Mic+18]. Every operation in the training pipeline is done natively in GPU using Torchvision [MR10]. These optimizations allow us to train the networks one order of magnitude faster than previous methods [Zho+18]. The input textures T tested in this chapter have, on average, 500 pixels in their larger dimension, for which our method can generate tiles of at most 1000 pixels. We use Photometric Stereo [Ike81] for computing the normals of fabric textures, and artist-generated normals for the other samples.

Figure 3.8: Crops of synthesized textures using our method. We observe that pixel-wise coherence between maps is preserved.



We tile the latent space at its earliest level ($F_l, l = 0$), as it provides the best quality results, as shown in Section 3.5.1. For identifying the tileability of textures, we use a search area S that spans a 20% of each spatial dimension of the textures. For all the results shown, $\gamma = 1$. We choose $c_{\min} = 100$ pixels, and $c_{\max}$ equal to the resolution of the whole input texture. This means that for the level of $c_{\max}$ only one texture is sampled. For each $c < c_{\max}$, we sample 3 different random crops. Typically, for the results shown in this chapter, a tileable texture stack is found at high-resolution crop sizes, $c \geq c_{\max} - 10$, thus needing to sample and evaluate less than thirty random crops before finding a satisfactory solution. This entire sampling procedure takes less than five minutes.

3.7 Experiments

In this section, we evaluate two of our main contributions: first, in Section 3.7.1 we study the impact of the design choices of the generator G to synthesize a multi-layer texture stack, and second, the quality of the tileable texture synthesis through several ablation studies. We study the inter-map consistency in Section 3.7.2, and the impact of the loss function in Section 3.7.3. Finally, we compare our method with methods on tileable texture synthesis (Section 3.8).

3.7.1 Generator G Design

One of the main challenges when synthesizing texture stacks using a single generator is to preserve the low-level details of each of the texture maps whilst maintaining the local spatial coherence between them; if this coherence is lost, renders that use the synthesized maps will show artifacts or look unrealistic. We propose two different variations of the single-map generative architecture G presented in [Zho+18], each of which makes different assumptions on how the synthesis should be learned taking into account the particular semantics and purpose of each map. A diagram of each proposed architecture is shown in Figure 3.10. For a fair evaluation, we follow the criteria that both networks must have approximately the same number of trainable parameters.

Our baseline, G_1 , treats the texture stack as a multiple-channel input image, and entangles every texture map in the same layers. It assumes that the maps in the stack share most of the structural information and, as such, there is no need to generate them separately. Thus, the last layer in the decoder outputs every texture map. Our proposed alternative architecture, G_2 , finds a shared representation of each texture map, but has a separate decoder for each of them. As such, the residual blocks are shared for all the texture stack, but each decoder can be optimized for the semantics and statistics of each particular map.

To quantitatively evaluate which architecture produces the highest quality output we compare the original texture with the generated one using standard metrics: SSIM, Si-FID, and LPIPS. The *Structural Similarity Index Measure (SSIM)* [Wan+04] is a perceptual-aware metric, working on the pixel space, that measures the similarity in structural information, and may be appropriate to evaluate synthesized textures. The Si-FID [SDM19] is a single image extension of the *Fréchet Inception Score (FID)* [Heu+17], which measures the difference in deep latent statistics between natural and artificially generated images. Finally, we use the *Learned Perceptual Image Patch Similarity (LPIPS)* [Zha+18] as a perceptual distance metric in deep image spaces. This metric is widely used for evaluating generative models [Kar+20b; Hua+18b; Cha+19; Alm+18; Mar+20].

The baseline G_1 shows more artifacts than G_2 , most likely due to the fact that the generation parts of the network are not fully separated. A quantitative evaluation is shown in Table 3.1, showing that G_2 outperforms G_1 from perceptual and statistical standpoints. Those metrics require the input and target images to have the same spatial dimensions. To obtain this, we crop the 50% center area of each generated stack, which doubles the dimensions of the inputs due to the adversarial expansion approach we follow, and compare them with the input textures. In summary, G_2 allows to generate textures that better preserve the properties of the input images without any additional computational cost and without requiring to modify the loss function, or the discriminator design.

Table 3.1: Quantitative comparison between our variations of the generator G . We show the average results on different metrics across different texture stacks, separated by maps. As shown, G_2 outperforms its baseline across metrics and texture maps. G_1 only yields better scores at the Si-FID metric on the normal map. Higher is better for SSIM, while lower is better for Si-FID and LPIPS

		SSIM [Wan+04] $\uparrow$	Si-FID [SDM19] $\downarrow$	LPIPS [Zha+18] $\downarrow$
G1	Albedo	0.2774	0.3462	0.4826
	Normals	0.2364	0.3636	0.4870
G2	Albedo	0.3123	0.3275	0.4377
	Normals	0.2921	0.4019	0.4340

3.7.2 Inter-map Consistency

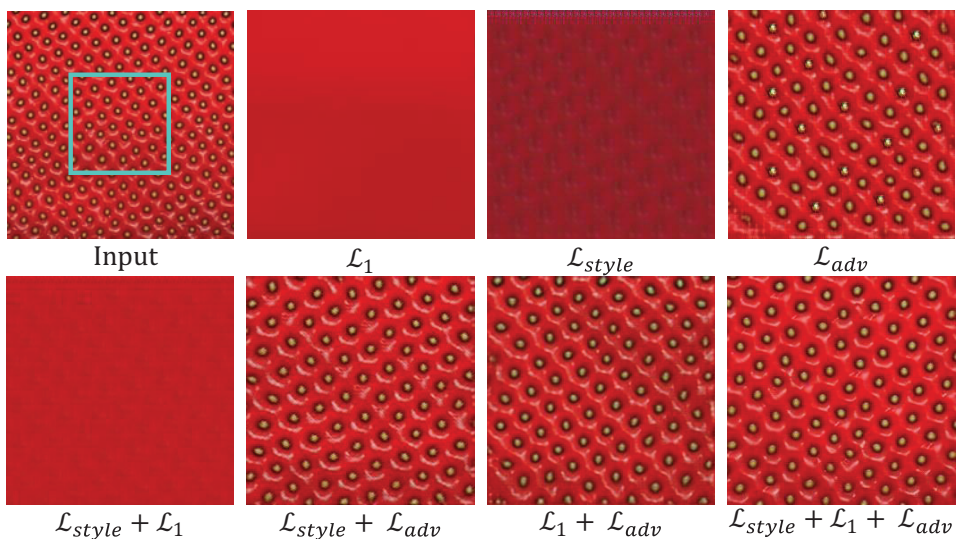
Whilst our generators G can generate high-quality tileable pairs of albedo and normal maps, there is no guarantee that those maps are pixel-wise coherent, as no part in the loss function explicitly accounts for this relationship. The L_{style} loss is computed separately for each map in the texture stack, which may generate non-coherent gradients. Furthermore, our architecture of choice G_2 separates the decoding of each map in different parts of its architecture, which may hinder the generation of spatially-coherent maps. Nevertheless, we show that this is not a problem in practice. Figure 3.8 shows crops of our synthesized texture maps. It can be seen that pixel-wise coherency between maps is preserved even for challenging geometric structures. We argue that the role of the discriminator is key to detect inter-layer inconsistencies by yielding lower probabilities for non-coherent maps. Even if separating the decoders may increase the risk of incoherent texture maps, this coherence is forced by the discriminator during training. To test this empirically, for the textures in Figure 3.8, we fed the discriminator with a crop of the original stack, a crop with the normals translated 5

px, and translated 100 px, for which we obtain average values of 0.99, 0.35, and 0.32, respectively. This suggests that, since during training, the discriminator receives the entire texture stack at once, it learns to identify spatial inconsistencies between maps.

3.7.3 Loss Function Ablation Study

As described in Section 3.4, the generator G is trained to minimize a loss function, which is comprised of adversarial, perceptual and pixel-wise components. Each of these have different impacts on the output synthesis. We hereby study the impact of each of these components.

Figure 3.9: Ablation study on the impact of the loss function on the quality of the synthesized textures. We evaluate each network using the same input, marked using a blue box. As shown, training G using the full loss function yields the best results. Each component is weighted by a λ , as specified in Section 3.6.

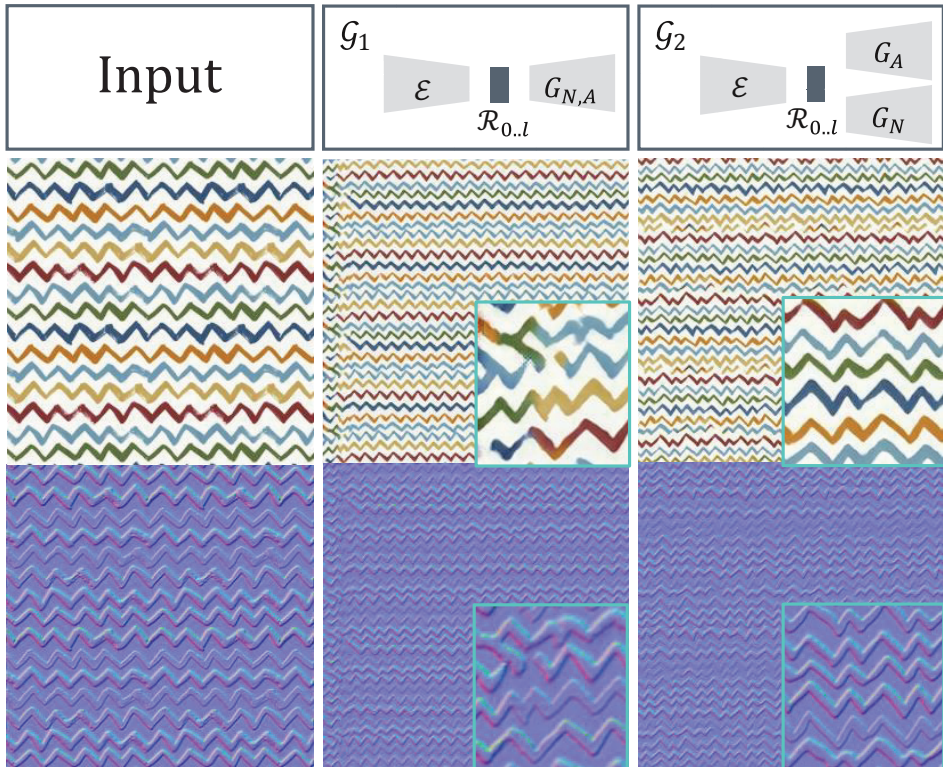


To better isolate the impact of each metric, we perform this study using a limited generator that only outputs albedo maps. As we show in Figure 3.9, the adversarial loss $\mathcal{L}_{adv}$ provides high-level semantic consistency, the $\mathcal{L}_1$ norm acts as a regularization method which removes artifacts, while the perceptual loss $\mathcal{L}_{style}$ adds additional details to fully represent the texture. Similar findings are reported in [Zho+18]. None of these loss functions yield compelling or artifact-free results on isolation, being the adversarial loss the most important factor of the global function.

3.8 Results and Comparisons

Most tileable texture synthesis methods have not shown results in synthesizing texture stacks. While adding this capability is reasonably easy for some *non parametric* methods [Li+20; Rod+19], others require major changes in their design. In particular, as we have shown in Section 3.7.1, expanding the architecture of deep learning methods for generating more than one map requires special attention to balance the map's interdependence with the network's capacity and expected quality. Therefore, in this section, in order to be able to compare our method with other works, we use a reduced version of our generator that only outputs a single albedo map.

Figure 3.10: A comparison of the results of our proposed architectures. Separating the decoders of the network for each map (G_2) outperforms the joint-decoder baseline (G_1) for both texture maps in the stacks. It generally allows for more varied albedo maps, with a more correct structure and style; as well as more accurate and sharper normal maps. The outputs in this figure are not tiled, for improved visibility of the artifacts created by G_1 . In blue, we show an up-sampled crop of the output textures.



Qualitative Analysis First, we compare with the Texture Stationarization algorithm proposed in [Mor+17] using examples taken from their own dataset, as their code is not publicly available. A key difference between both methods is that, while their method aims to maximize the stationarity properties of the textures using external metrics, our method learns them from the texture itself, using a self-supervised approach. This results in our method better preserving the content of the original input. As shown in Figure 3.11, our method shows compelling results for the kind of textures shown in their paper. In the *fence* example, our methods better preserves the vertical straight lines. For the *wall* example, both methods shows compelling results, capturing a different repetition pattern.

As Moritz *et al.*'s dataset contains mainly textures of human-made environments, we gather a different set of images of greater variety in their regularity and content. Figure 3.12 shows a comprehensive comparison with the other methods. The work by Li *et al.* [Li+20], based on *Graph-Cuts*, works reasonably well if the transformation can be done locally at the seams but fails when the required changes are global, as happens in the *basket*. The *Repeated Pattern Detection* algorithm proposed by Rodriguez-Pardo *et al.* [Rod+19] is not able to handle many of these challenging cases, which do not represent grid-like textures, as in the *bananas* or *stars*. The Periodic Spatial GAN (PSGAN) proposed by Bermann *et al.* [BJV17] generates artifacts, not maintaining the content of the texture, as in the *daisies* or the *hieroglyph* examples. A similar effect is observable in the method of [Nik+21], which uses self-organizing representations through neural cellular automata. The output of the method is seamless but the integrity of the original texture is not preserved in many cases, for example, in the *hieroglyph* or the *stars*.

Our method favors keeping high-level and bigger semantic structures of the textures, resulting in larger samples with more variety. The modification applied to the texture is the minimum required to make it seamlessly tileable. It provides high-quality outputs for regular textures with uneven illumination and perspective distortion, such as the *basket*, textures which are greatly irregular, such as the *bananas*, and stochastic ones, like the *strawberry*. While non-parametric synthesis methods are typically computationally cheap, obtaining results in less than one minute [Li+20; Rod+19; DH19], at the cost of quality and generality, parametric methods are more expensive and slower. Our method is competitive in computational cost when compared to other parametric synthesis methods. Our entire training and sampling process takes less than 45 minutes. In comparison, PSGAN [BJV17] needs 4 hours to train, the work by Zhou *et al.* [Zho+18] requires 6 hours, and the method of Niklasson *et al.* [Nik+21] takes 35 minutes to generate textures limited to 200×200 pixels. On the contrary, our method is not limited by the size of the input texture and can synthesize tiles of any size thanks to the fully-convolutional architecture, hardware being the only limiting factor.

Figure 3.11: A comparison with the work on Texture Stationarization by Moritz *et al.* [Mor+17], using textures from their dataset. The results are tiled 2×2 times to help visualization.

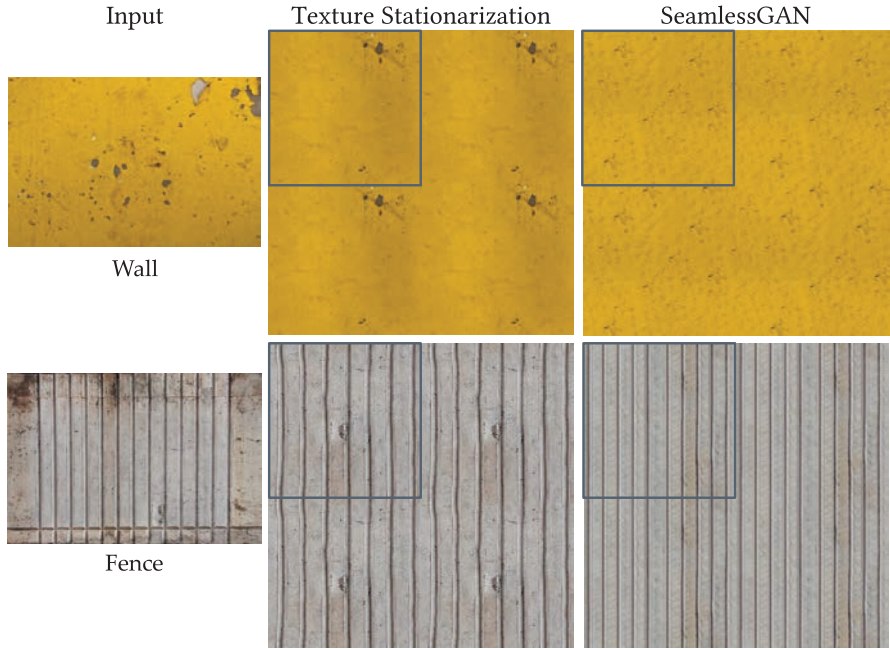


Table 3.2: Quantitative comparison between different methods. We show the average results on different perceptual metrics across a variety of textures. As shown, SeamlessGAN consistently outperforms its counterparts in every studied metric. We tile the outputs until they match the spatial resolution of the input examples. Higher is better for SSIM, while lower is better for Si-FID and LPIPS. We use a color code to highlight **best** and **worst** cases.

	SSIM [Wan+04] $\uparrow$	Si-FID [SDM19] $\downarrow$	LPIPS [Zha+18] $\downarrow$
Deloit <i>et al.</i> [DH19]	0.1424	1.3471	0.6207
Li <i>et al.</i> [Li+20]	0.2086	0.9529	0.5818
Moritz <i>et al.</i> [Mor+17]	0.1968	0.7620	0.5171
Rodriguez-Pardo <i>et al.</i> [Rod+19]	0.2144	1.2958	0.5137
Bergmann <i>et al.</i> [BJV17]	0.1723	1.4355	0.5624
Niklasson <i>et al.</i> [Nik+21]	0.1753	1.3171	0.5328
SeamlessGAN	0.2341	0.6311	0.4792

Quantitative Comparison Following a similar evaluation scheme as proposed in Section 3.7.1 for measuring the differences between input and synthesized images, a quantitative evaluation is shown in Table 3.2. We use Moritz’s dataset for a fair comparison with every method. The metrics we use for this experiment require that the input and synthesized images have the same resolution. To achieve this, and, in order to account for artifacts in the borders of the generated textures, we tile the synthesized images until they cover the same resolution as their corresponding inputs.

Interestingly, methods based on patches [RB19; Mor+17; Li+20] obtain better SSIM and Si-FID scores than previous deep learning-based methods [BJV17; Nik+21]. This difference is not seen in the LPIPS metric. This suggests that patch-based methods preserve better the structure of the input textures than previous neural parametric models. Our method, by combining a variety of loss functions, allows for better preservation of the style and semantic content of the generated textures. Furthermore, our latent space manipulation algorithm allows for seamless borders between tiles, outperforming both previous non-parametric and parametric methods in perceptual and structural similarity. The magnitude of the Si-FID metric varies significantly between the values obtained in Table 3.1 and Table 3.2, which indicates that this metric may be overly sensitive to the global statistics of the dataset.

Limitations and Discussion Our method is inherently limited by the capabilities of the adversarial expansion technique to learn the implicit structure of the given texture. That is, if the input does not show enough regularity, the adversarial expansion fails, as shown in Figure 3.13. It is interesting to see that the synthesis of the *dotted* texture fails to reproduce the larger dots, as they are very scarce. However, the remaining structure is very well represented.

Tiling single texture maps, even if they contain seamless borders, may generate perceptible repetitions. This is an intrinsic limitation of any single sample tileable texture synthesis. Approaches such as *Wang Tiles* [Wan+04] can tackle these limitations, but have additional disadvantages like increased memory and run-time consumption, rendering them less practical for real-time applications. Alternatively, procedural methods [Gue+20] are the most effective way to generate material samples preserving textural properties at multiple scales; however, present several limitations in the range of materials that can be generated to make them fully usable.

3.9 Conclusions and Future Work

In this chapter, we have proposed a deep parametric texture synthesis framework capable of synthesizing textures into tileable single-tiles, by combining recent advances in deep texture synthesis, adversarial neural networks, and latent spaces manipulation. Our results show that our method can

Figure 3.12: Comparison with previous methods. On the left column, we show the input textures. From left to right, the synthesized results of Histogram Blending, by Deloit *et al.* [DH19], the GraphCuts algorithm by Li *et al.* [Li+20], Repeated Pattern Detection by Rodriguez-Pardo *et al.* [Rod+19], PSGAN by Bergmann *et al.* [BJV17], Self-Organizing Textures by Niklasson *et al.* [Nik+21]; and ours. Outputs are tiled a similar number of times (at least twice in each dimension) for better visualization.

Our method generally captures better the overall structure of the texture, while providing seamless and semantically coherent borders, for enhancing tileability. From top to bottom, we sample $c = 4, 12, 5, 17, 13, 1, 19$ and 43 input crops before obtaining a tileable texture.

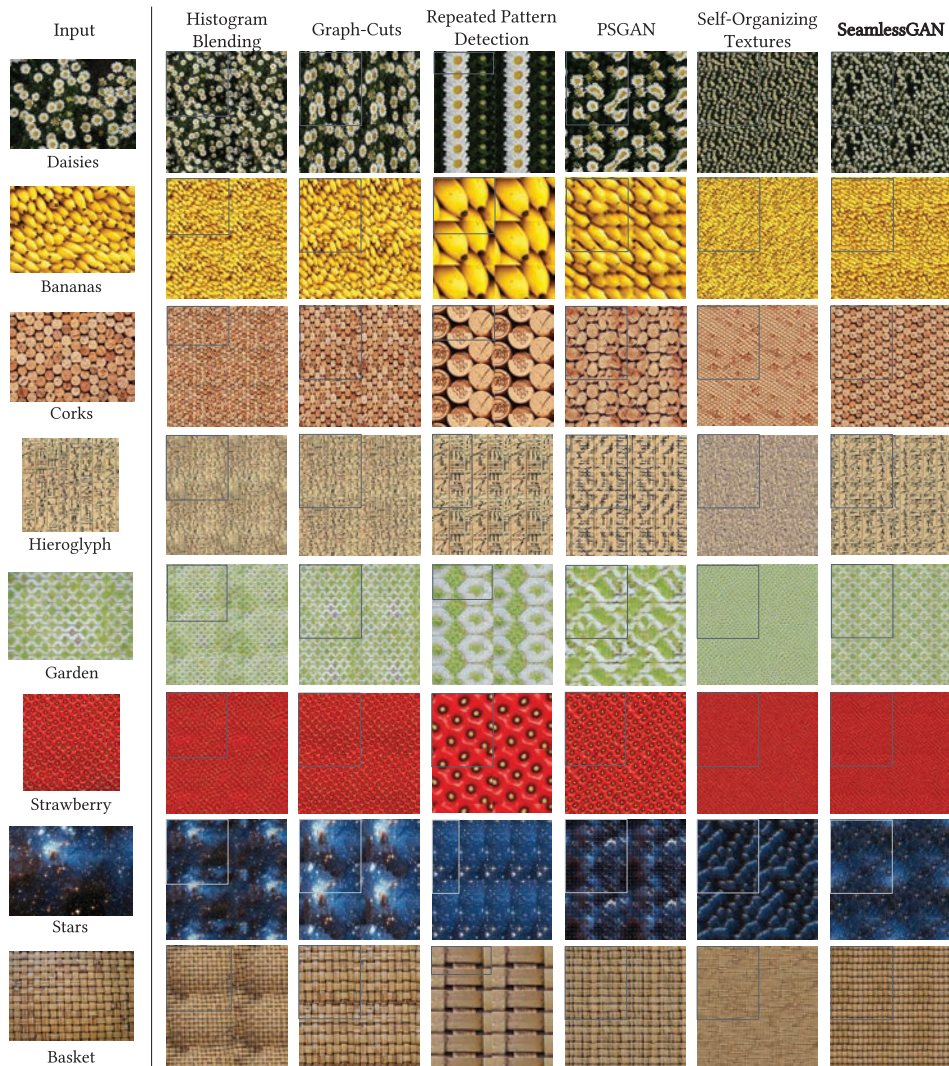
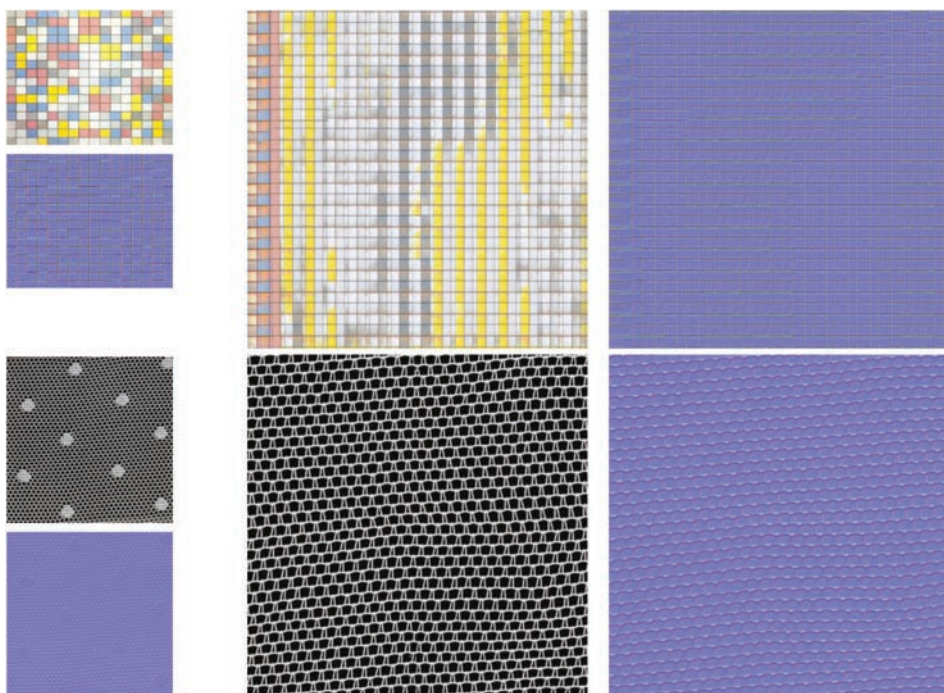


Figure 3.13: A limitation of our texture synthesis algorithm: Left: Input texture stacks; right: synthe-sized stacks. The network fails to replicate the pattern if the occurrence is not frequent enough, as we can see in the base color of these two examples. On the other hand, the synthesized normal map is consistent as its repetitive structure is seen frequently by the network.



generate visually pleasing results for images with different levels of regularity and homogeneity. This work is the first method capable of exploiting properties of deep latent spaces within neural networks for generating seamless textures, and opens the opportunity for end-to-end tileable texture synthesis methods without the need for manual input. Comparisons with previous state-of-the-art methods show that our method provides results which better maintain the semantic properties of the textures, while being able to synthesize multiple maps at the same time.

Our method can be improved in several ways. First, the adversarial expansion framework, while powerful, it has some potential pitfalls that hinder its widespread applicability. The same neural architecture is used for every texture but, as discussed, different choices on the architecture make different assumptions on the nature of the textures. We have proposed a generic architecture that works well for many examples, but recent advances in Neural Architecture Search [Mel+21]

may provide better priors on the optimal neural architecture to use for each sample. Furthermore, each texture synthesis network is trained from scratch. This is not only computationally costly, but learning to synthesize one texture may help in the synthesis of other textures, as shown by [Liu+20]. Fine-tuning pre-trained models for generating new textures may provide cheaper syntheses. Besides, our discriminator, while capable of detecting local artifacts, provides little control for separating such artifacts and global semantic errors. Recent work on image synthesis may provide guidance onto designing better discriminative models [SSK20], training procedures [Yun+19; Sin+21], image parametrizations [Wan+20a; HZ21; Nik+21] or perceptual loss functions [Hei+21].

Second, we proposed a synthesis solution based on manipulating latent spaces within the generative model, but explicitly training the network to generate tileable textures may provide better results than our approach. Besides, our sampling procedure could be extended for better selection of textures, by comparing histograms of the activations of the discriminator on the selected central area, instead of simply comparing minimum values.

Finally, our method has the advantage of being fully automatic, however, pre-processing the texture images so they are more easily tileable can help the synthesis process. For example, automatically rotating the textures so their repeating patterns are aligned with the axes was studied by [Rod+19]. Powerful methods for artifact [Dek+15] and distortion [Li+19] removal could be applied as a pre-processing operation to the input textures before training the generative models, or as additional components to their loss function for improving tileability or homogeneity.

3.A Additional Details

Training details: We use PyTorch [Pas+19] as our learning framework, and Adam [KB15] as our optimization algorithm. For it, we use an initial learning rate of $\alpha = 0.0002$, which is divided by 5, after iterations 30000 and 40000. The networks are trained for a total of 50000 iterations, using a batch size of 1. The momentum parameters are kept to the default values of $\beta_1=0.9$ and $\beta_2=0.999$, and $\epsilon=10^{-8}$. We additionally apply an L_2 regularization of 10^{-5} to the optimization of both networks, which may help obtain better synthesized images [Kur+19]. This configuration is the same for the training of both the discriminator and the generator. In contrast to other work in unsupervised image generation [Zhu+17a], both networks are updated after each iteration. The models are trained and evaluated using a single NVIDIA GeForce GTX 1080Ti, leveraging half-precision training and automatic gradient scaling [Mic+18]. Evaluation is done in half precision for faster inference and reduced memory consumption. Using reduced numeric precision does not yield worse synthesized textures.

Network architecture: We follow closely the architectures defined in [Zho+18]. For the **Generator** G , we use replication padding as the padding method for the entire architecture, ReLU non-linearities, and Instance Normalization [UVL16] before each non-linearity. This network receives textures of $3 \times k \times k$ pixels, which are first processed by an encoder E , which transforms it to a $256 \times \frac{k}{4} \times \frac{k}{4}$ latent vector x_0 . Following standard practice on residual convolutional networks [He+16], the first layer of E is comprised by $64 \ 7 \times 7$ convolutional filters. The rest of the network, unless stated otherwise, uses standard 3×3 kernels. This latent vector is further processed by 5 residual blocks $x_i \leftarrow R_i(x_{i-1}) + x_{i-1}$, which maintain the latent vector dimensions. Then, a decoder G takes x_5 and, using three convolutional layers with transposed convolutions [LSD15], followed by ReLU operations, it returns the estimated texture, with $2k \times 2k$ resolution. As in the encoder, the last layer of G contains $64 \ 7 \times 7$ convolutional filters. There are as many decoders G as individual maps in the texture stack. When tiling the latent space x_0 , this doubles its spatial resolution, thus becoming a $256 \times \frac{k}{2} \times \frac{k}{2}$ latent vector. As such, once processed by the residual blocks and the decoders, the network outputs a $4k \times 4k$ texture stack.

For the **discriminator** D , we simply follow a *PatchGAN* architecture [Zhu+17a; Iso+17; Led+17; Zho+18], which outputs local estimations of the probability of the input image being real or generated. We use 4 blocks of layers, each comprised of, in order, a convolutional layer with 4×4 kernels, an Instance Normalization [UVL16] layer, and a Leaky ReLU non-linearity [MHN13]. We use a stride of 2 in the first and last layers to decrease the dimensions of the latent representations. The first block of layers contains 64 trainable kernels, which are doubled after each block. The last layer thus contains 512 trainable filters. In the last layer, we remove the normalization operation and substitute the non-linearity by a Sigmoid operation, to transform the output into the (0, 1) range. The discriminator receives $2k \times 2k$ resolution stacks, and outputs a $1 \times \frac{k}{2} \times \frac{k}{2}$ resolution estimation map.

3.B Additional Comparisons, Results and Ablations

In this section, we outline the implementation details used for our comparison with other tileable texture synthesis methods, which included Histogram-Preserving Blending by Deliot and Heitz [DH19], Graph-Cuts synthesis by Li *et al.* [Li+20], Repeated Pattern Detection by Rodriguez-Pardo *et al.* [Rod+19], Periodic Spatial GAN by Bergmann *et al.* [BJV17] and Self-Organizing Textures by Niklasson *et al.* [Nik+21]. We also provide further results in Figure 3.15 and an ablation study on the size of the model architectures in Figure 3.14.

Histogram-Preserving Blending We use the implementation provided by the authors. The only hyperparameter to be changed is the blending border size, which is set to the default value of 33% for all the textures.

Graph-Cuts synthesis To allow for comparison between single-map texture synthesis methods, we run this method to output only the albedo map, using the default values of $\lambda_d = 1$, a ratio of 0.64, a crop ratio of 0.6 and a margin size of 72 pixels. The images are not rescaled to any input or target size, instead, we use their original resolution.

Repeated Pattern Detection This paper proposed the use of some preprocessing methods for improving the regularity of the input images, which were not used for our comparison. We use their default hyperparameter values, including $\delta = 0.65$, $\lambda = 0.8$, and the deep perceptual loss weighting proposed in [GEB15a].

Periodic Spatial GAN This method can handle both single-image and multiple image datasets. We are interested in the former, and, as such, each GAN is trained using only one texture sample. We train their model using Theano [AI+16], and use their default configuration of a learning rate of 0.0002, a $\beta_1 = 0.5$ as the momentum term for the optimization algorithm, a batch size of 25, and train their network for 10 epochs, each comprised of 25000 training steps. The network parameters and sizes are exactly as defined in the paper.

Self-Organizing Textures This method is computationally expensive and, as such, it requires that the input textures are rescaled to 128×128 pixels. We train their method using Tensorflow [Aba+16] for 5000 iterations. We use the default training configuration, with a starting learning rate of 0.002, which is then reduced by half at iterations 2000 and 4000. We do not modify the image parametrization design or loss function in any way.

Figure 3.14: An ablation study on the influence of the number of layers in each model in the quality of the generated outputs. We evaluate each network using the same input, marked using a blue box. We compare the influence of using discriminators with $L \in \{3, 4, 5\}$ layers, marked as D_L and generators with $L \in \{4, 5, 6\}$ residual blocks, marked as G_L . We show the configuration we use for every experiment in this chapter using a green box. On the bottom of each image, we show the training time for 50000 iterations. As shown, using a discriminator with 3 layers yields repetitive large-scale patterns, due to its reduced receptive field. Using $D_L = 5$ yields marginal improvements with respect to $D_L = 4$, with the cost of increased runtimes. In terms of G_L , we find that there are no significant visual differences between using 5 or 6 residual blocks, but we favor the former due to its reduced training time.

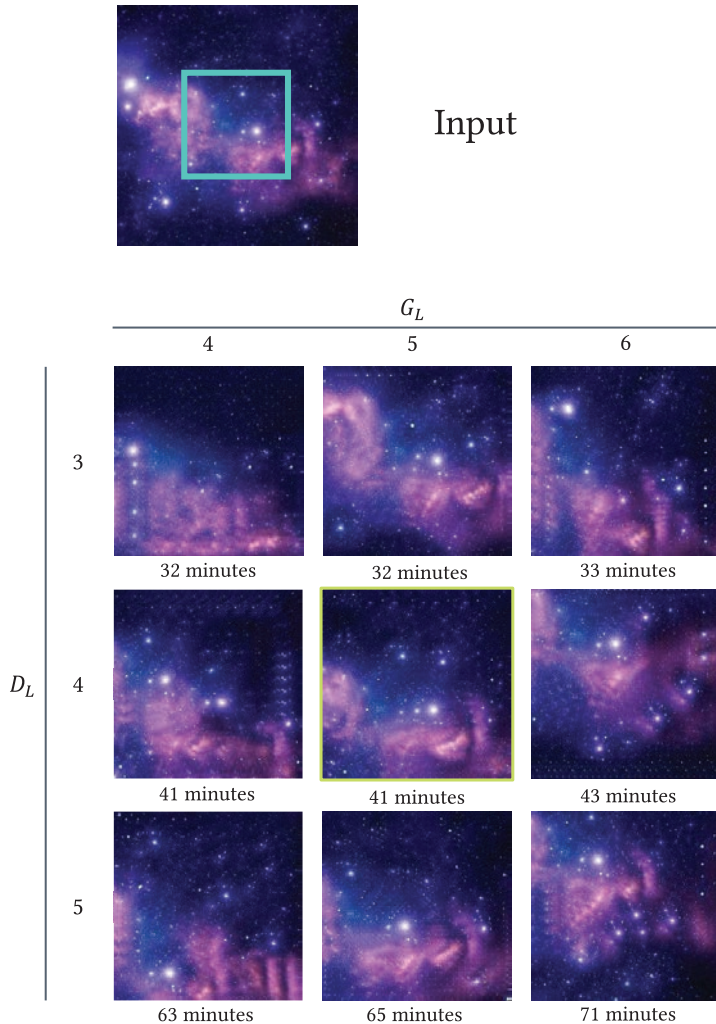


Figure 3.15: More examples on the capabilities of SeamlessGAN on generating multiple tileable textures from a given input. On the left, we show the input sample T and four crops used for generating tileable maps. On the middle column, the outputs of the generative model are shown. On their right, the same textures are tiled 2×2 times for visualization purposes.

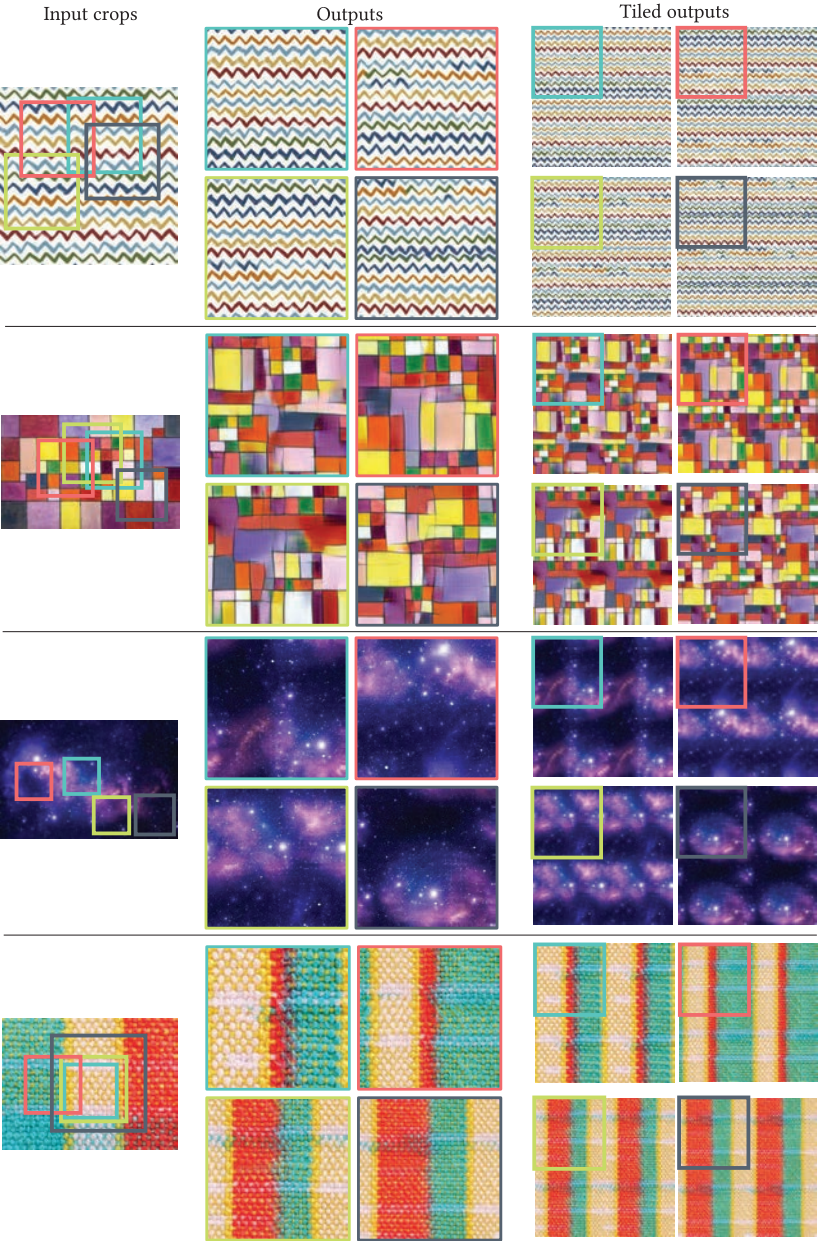


Figure 3.16: Additional comparison with previous methods. From left to right, top to bottom: Input texture, the results of Histogram Blending, by Deloit *et al.* [DH19], the GraphCuts algorithm by Li *et al.* [Li+20], Texture Stationarization by Moritz *et al.* [Mor+17], Repeated Pattern Detection by Rodriguez-Pardo *et al.* [Rod+19], PSGAN by Bergmann *et al.* [BJV17], Self-Organizing Textures by Niklasson *et al.* [Nik+21]; and ours. Outputs are tiled a similar number of times (at least twice in each dimension) for better visualization. Our method generally captures better the overall structure of the texture, while providing seamless and semantically coherent borders, for enhancing tileability. From left to right, we sample $c = 27, 36, 29, 1, 17$, and 7 input crops before obtaining a tileable texture.



CHAPTER 4. NEURAL FIELDS FOR BTF ENCODING AND TRANSFER

Figure 4.1: On the left, renders generated with NeuBTF materials. On their right, a slice of the input training BTF and the guidance image onto which we propagate the BTF measurements.



Neural material representations are becoming a popular way to represent materials for rendering. They are more expressive than analytic models and occupy less memory than tabulated BTFs. However, existing neural materials are immutable, meaning that their output for a certain query of UVs, camera, and light vector is fixed once they are trained. While this is practical when there is no need to edit the material, it can become very limiting when the fragment of the material used for training is too small or not tileable, which frequently happens when the material has been captured with a gonio-reflector. In this chapter, we propose a novel neural material representation that jointly tackles the problems of BTF compression, tiling, and extrapolation. At test time, our method uses a guidance image as input to condition the neural BTF to the structural features of this input image. Then, the neural BTF can be queried as a regular BTF using UVs, camera, and light vectors. Every component in our framework is purposefully designed to maximize BTF encoding quality at minimal parameter count and computational complexity, achieving competitive compression rates compared with previous work. We demonstrate the results of our method on a variety of synthetic and captured materials, showing its generality and capacity to learn to represent many optical properties. The contributions in this chapter led to the following publication, currently under review:



"NeuBTF: Neural Fields for BTF Encoding and Transfer"

Carlos Rodriguez-Pardo, Konstantinos Kazatzis, Jorge Lopez-Moreno, Elena Garces

Under Review (2023)

4.1 Introduction

A common approach to modeling real-world spatially-varying materials in computer graphics is through the use of Bidirectional Texture Functions (BTFs). This type of representation models the dense optical response of the material, and is more general than analytic representations such as a micro-facet SVBRDF. However, BTFs typically occupy large amounts of memory. Recently, neural material representations are being proposed as an learning-based alternative to tabulated BTFs, providing a more compact solution while keeping the flexibility and generality of BTFs.

Creating digital representations of real material samples requires using an optical capture device, such as a gonioreflectometer [Gar+23], a smartphone [Des+18; Des+19; Guo+20b; Hen+21], or a flatbed scanner [Rod+23a]. During the process, several choices must be made. First, it is important to select a patch of the material that contains enough spatial variability. Second, a process automatic or manual must be found to produce a tileable material that can be used to create seamless 3D renders. Finally, resources must be allocated for storage as needed. Making these choices when dealing with implicit or tabulated representations, such as in BTFs or neural materials, is particularly crucial. Once these representations are trained or captured, they cannot be easily modified and it is only possible to query them using the UVs, light, and camera vectors.

In this chapter, we propose a novel neural material representation that addresses these issues. Unlike existing neural approaches that are immutable once trained [Rai+19; Kuz+21; Rai+20; Kuz+22], our model can be queried a test time with a guidance image that conditions the neural BTF to the structure provided by the guidance image. Our approach resembles synthesis by example and procedural processes, and can be used to extrapolate BTFs to large material samples, as well as to easily create tileable ones. Furthermore, our method achieve better compression rates than previous work on neural BTF representations.

To achieve this, we present a novel deep learning method that works at two steps. In the first step, we condition the neural BTF using a *guidance image*

as input. To this end, we use an *autoencoder* that outputs a high-dimensional latent representation of the material, a *neural texture*, which jointly encodes reflectance and structural properties. In the second step, the UV position of the latent representation, along with the camera and light vectors, are decoded by a fully-convolutional sinusoidal decoder [Sit+20], a neural *renderer* to obtain the RGB values. Using a single BTF as input, we train the network end-to-end using a custom training procedure, loss function, and data augmentation policy. This policy, inspired by recent work on attribute transfer [RG21] (Chapter 2), allows the autoencoder to encode the relationship between structural features and reflectance, enabling the propagation of the BTF to novel input guidances. Once trained, the novel input guidances may come from the same material, a different material, or a structural pattern. An input guidance of the same material can be used to extrapolate the BTF to larger samples or create tileable BTFs, provided the input guidance is tileable. If the input guidance is a structural pattern, the local features can be used to synthesize novel materials.

In summary, we propose the following contributions:

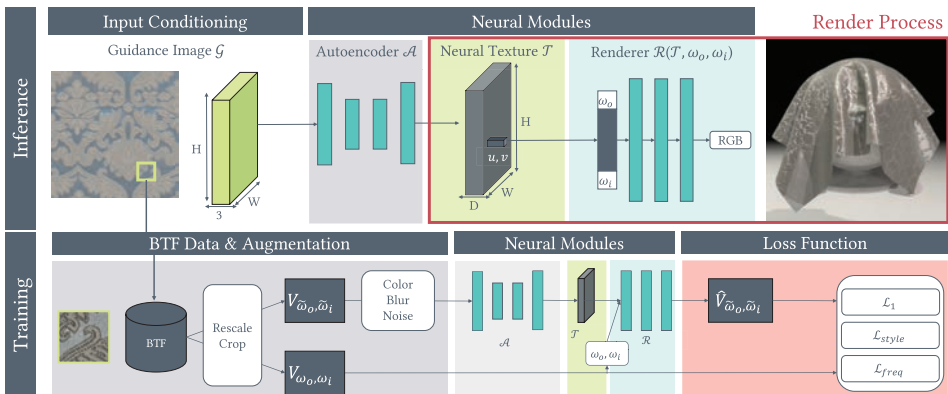
- The first neural BTF representation with conditional input that can be used to extrapolate BTF measurements, easily create tileable BTFs, and synthesize novel materials.
- We show how to leverage our system for rendering large-scale and tileable neural BTF generation using measurements captured with small portions of the material.
- We demonstrate that our method works with synthetic and captured materials of diverse optical properties, including colored specular or anisotropy.

4.2 Related Work

An accurate method for representing the optical properties of materials is through Bidirectional Texture Functions (BTFs) [Dan01]. BTFs are 6D functions that characterize all possible combinations of incoming and outgoing light and camera directions for the 2D spatial extent of a material. Although they are successful in representing materials, they have a major drawback in terms of memory requirements. Therefore, BTF compression has been a major research topic [FH08]. Non-neural approaches used dimensionality reduction techniques such as Principal Component Analysis (PCA) [Kou+03; MMK03; WGK14], vector quantization [HFM10], or clustering [Ton+02]. However, these approaches were recently surpassed by neural models [HS06] due to their flexibility and superior capacity to learn non-linear functions.

Neural BTFs Rainer *et al.* [Rai+19] proposed the first method to use deep autoencoders to compress BTFs, surpassing PCA [WGK14] on captured BTFs. However, this approach required training a single neural network per material. To address this limitation, a later work by the same authors [Rai+20] proposed a generalization of this idea in which a single network was able to generalize to a variety of materials. Although these methods were very effective for compressing flat materials, they had some limitations when it came to modeling materials with volume. In their work, Kuznetsov *et al.* [Kuz+21] improved the quality of neural materials by introducing a neural offset module that captures parallax effects. Further, their method also allowed for level-of-detail through MIP mapping by training a multi-resolution neural representation. However, grazing angles and silhouette effects remained a challenge for this approach. In a subsequent work, Kuznetsov *et al.* [Kuz+22] explicitly trained the network using queries that span surface curvatures, effectively handling these cases. Representing fur, fabrics, and grass with neural reflectance fields was explored by Baatz *et al.* [Baa+22] who proposed a representation that jointly models reflectance and geometry

Figure 4.2: An overview of our neural BTF inference and training processes. **Top-Inference:** Using a guidance image $g \in \mathbb{R}^{H \times W \times 3}$, we use our trained autoencoder to generate the Neural Texture $A(G)=T_g \in \mathbb{R}^{H \times W \times D}$, which preserves the spatial resolution of the input image but represents a higher-dimensional learned representation. This Neural Texture T_g , along with the trained renderer R , can be queried as a regular BTF, using UVs, and target camera and light positions for regular rendering. **Bottom-Train:** During training, we randomly select an input view $V_{\hat{\omega}_0, \hat{\omega}_i}$, and a target V_{ω_0, ω_i} . To both views, we apply random rescale and cropping. Then, only to $V_{\hat{\omega}_0, \hat{\omega}_i}$, we randomly apply hue variations, gaussian blur, and noise, and feed it to the *autoencoder*, which returns a 2D *latent representation* of the material. A fully-convolutional *decoder* with sinusoidal activations receives both this latent space and the target ω_o, ω_i camera and light angles, and estimates $\hat{V}_{\omega_o, \omega_i}$. This output is compared with V_{ω_o, ω_i} using a multifaceted loss function.



All of these approaches share the idea of querying neural material using UVs, camera, and lighting vectors, but do not provide any functionality for modifying the material once the network is trained. In contrast, our approach can take a guidance image as input, which conditions the output to generate material variability.

Material Synthesis and Tiling Texture synthesis is a long-standing problem in the field of computer graphics. The goal is to reconstruct a larger image given a small sample, leveraging the structural content and internal statistics of the input image. This concept has been used for synthesizing single images, BTFs, and full material models. For images, the most common strategies include PatchMatch [Dia+15], texture transport [AWL15], point processes [Gue+20; LH06], or neural networks [EM17; Zho+18; FAW19; RG22]. BTF synthesis, however, has received less attention. Steinhausen et al. [Ste+15a; Ste+15b] extrapolated

BTF captures to larger material samples using non-neural texture synthesis methods. For full materials, Li *et al.* [LPG19] captured the appearance of materials by first estimating their BRDF and then synthesizing the high-resolution micro-structure from a dataset of measured SVBRDFs. Nagano *et al.* [Nag+15] measured microscopic patches of the skin and used a convolutional filter to propagate the measurements to a spatially-varying texture. Deschaintre *et al.* [DDB20] used an autoencoder to propagate SVBRDFs to large material samples. Also recently, Rodriguez-Pardo and Garces [RG21] propagated any kind of visual attribute having a single image as guidance. Their approach shares some similarity to ours, although they transfer 2D image attributes, while we transfer the full BTF.

Procedural models [Hu+22b; Zho+23; Zho+22] are nowadays very successful for generating tileable materials. Thanks to the use of a tileable template, these methods adjust the generated image to the features available in the template. As we show, our approach can also work with a binary template as input. However, guaranteeing predictable outputs given this kind of input is out of the scope of our technique, which can transfer BTF measurements having as input a guidance image of the same material.

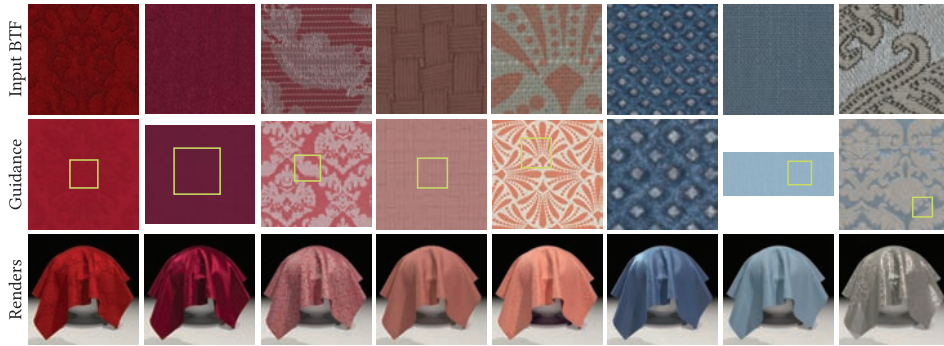
Other Neural Representations in Rendering Limited to BRDFs, neural networks trained with adaptive angular sampling have been explored to enable importance sampling [Szt+21], needed for Monte Carlo integration. Deep latent representations also allow for BRDF editions. For example, Hu *et al.* [Hu+20] demonstrate that autoencoders can outperform classic PCA for the purpose of editing. Other applications of neural encodings in rendering are numerous. For instance, they have been used for scene prefiltering [BSK23], where geometry and materials are simplified to accommodate the LoD of the scene using a voxel-based representation and trained latent encodings.

For anisotropic microfacets, Gauthier *et al.* [Gau+22] propose a cascaded architecture able to adjust the material parameters to the MIP mapping level. Encoding light transport using neural networks for real-time global illumination has also been explored [Rai+22; GMX22], showcasing promising results.

4.3 Method

We present an overview of our approach in Figure 4.2, where we show our inference and training pipelines. Our goal is twofold: First, find a compact representation for a BTF through the use of neural networks. Second, enable the extrapolation of the BTF according to guidance images used as input. In Section 4.3.1 we describe our inference pipeline and neural network, and in Section 4.3.2 our training process. Section 4.4 contains specific implementation and design details of the neural networks.

Figure 4.3: Some tileable neural materials achieved with our method. On the top row, we show a slice of the BTF used to train a NeuBTF representation. With the tileable guidance images shown on the middle row, we propagate the neural texture using our autoencoders. These neural textures can be rendered to generate realistic images (bottom). We highlight the training dataset surface area as a green inset.



4.3.1 Inference

Our neural network is composed of three modules: an *autoencoder* A , a *neural texture* $T \in \mathbb{R}^{H \times W \times D}$, and a *renderer* R . The renderer $R(T(u, v), \omega_o, \omega_i) = \text{RGB}$ takes as input the feature vector at the (u, v) coordinates of the neural texture T , the view ω_o and light ω_i positions, and returns an RGB value. R acts as a conventional BTF and can be used as such in any render engine. An input *guidance* image, $G \in \mathbb{R}^{H \times W \times 3}$, is used during inference to condition the generation of the *neural texture* T . This conditioning allows us to propagate the

learned reflectance to novel guidance images that can be: a larger sample of the same material, a different material, or a structural image. In the simpler case, the guidance image comes from the BTF used for training, and our process is equivalent to previous work [Rai+19; Kuz+21].

The autoencoder A takes the guidance image $G \in \mathbb{R}^{H \times W \times 3}$ as input and outputs a neural texture $T \in \mathbb{R}^{H \times W \times D}$ with the same size $H \times W$ as the input guidance image but with more latent dimensions D . As a result, each pixel in the guidance image has a higher-dimensional neural representation in T . Because it is trained without explicit supervision, this latent representation can capture the reflectance and structural patterns automatically. Kuznetsov *et al.* [Kuz+21] also used a latent neural representation of the material, however, lacking the initial autoencoder their approach cannot synthesize novel BTFs without retraining, while our conditioning module allow us to generate novel BTFs during test time. The autoencoder and the renderer are neural networks trained jointly, using an end-to-end image-to-image approach we describe below.

4.3.2 Training

Figure 4.2 (bottom) illustrates our training process. It has two objectives: First, equivalent to regular BTF encoding, we aim to find the mapping between camera direction ω_c , light direction ω_l , and output slices of the BTF: $(\omega_c, \omega_l) \rightarrow V_{\omega_c, \omega_l} \in \text{BTF}$. Second, we aim to condition the synthesis process with an input guidance image, G .

To this end, for training, we feed the model with images, $V_{\hat{\omega}_0, \hat{\omega}_1}$ which are randomly sampled from the BTF, and are subject to additional data augmentation processes.

This extensive augmentation process guarantees invariance to different input variations during test time, like camera or illumination conditions, while keeps consistency of the outputs.

Loss Function Design Our loss function, that compares ground truth slices V_{ω_0, ω_1} with generated ones $R(A(f(V_{\hat{\omega}_0, \hat{\omega}_1})), (\omega_c, \omega_l))$, is a weighted sum of three terms: a pixel-wise loss, a style loss, and a frequency loss,

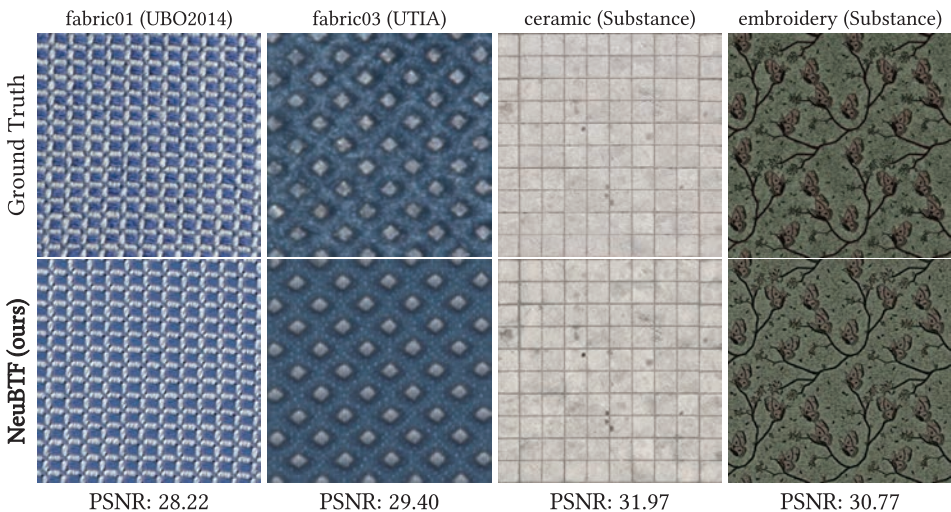
$$\mathcal{L} = \lambda_{L_1} \mathcal{L}_1 + \lambda_{style} \mathcal{L}_{style} + \lambda_{freq} \mathcal{L}_{freq} \quad (4.1)$$

The main driver of our loss is the pixel-wise norm L_1 . L_1 produces sharper results than higher-order alternatives, such as L_2 [Iso+17; RG21]. Following [Kuz+21], we apply a $\log(x + 1)$ compression to improve the model results on high dynamic range. This compression is only done to the pixel-wise component of the loss function. Inspired by recent work on texture synthesis, capture

and transfer [Mar+20; AWL13; RG22; Rod+23a; Zho+22], we introduce a L_{style} loss to help the model generate higher quality and sharper results. Further, to mitigate the spectral bias of convolutional neural networks and help ameliorate the results further, we also introduce a *focal frequency loss* into our learning framework [Jia+21]. This combination of loss functions proves effective for our problem, without the need for complex adversarial losses which could reduce efficiency or destabilize training.

Data Augmentation We train our models using a comprehensive data augmentation policy aimed at achieving high quality reflectance propagation, increasing performance and generalization, and allowing for generation of multiple resolution materials at test time. We build upon recent work on material transfer [RG21] and use images of the material taken under different illumination and viewing conditions as inputs to our autoencoder. This helps it generalize to novel capture setups, which allows for multiple applications we describe on Section 4.6. In particular, we use every image available on the input BTF, selected uniformly at random for each element in each batch during training. As in [Tex+20b; RG21; Rod+23a], we also use random rescaling, which helps the model generalize to new scales, and, and build neural materials of multiple resolutions at test time, as we describe on Section 4.6.3 inspired by recent work on image synthesis [RG21; Tex+20b; RG22], we use random cropping, which helps generalization by effectively increasing the dataset size. Finally, we extend the color augmentation policy in [RG21] with

Figure 4.4: Qualitative results on a variety of BTFs, from different sources. From left to right, we show results on the UBO [WGK14] BTF dataset, the UTIA [Hai12] BTF dataset and two synthetic materials rendered from Substance SVBRDFs.



random hue changes across the entire color wheel, and introduce random Gaussian noise and blurs to the input images, to help it generalize further, as proposed in [Rod+23a] (Chapter 5).

4.4 Implementation details

Our model size, function weighting, optimizer and training configuration, and data augmentation hyperparameters were selected using a combination of manual tuning and Bayesian hyperparameter optimization using *Weights and Biases* [Bie20].

Autoencoder For the *autoencoder*, we use a lightweight U-Net [RFB15] with a few modifications to tailor it for our problem. Inspired by recent work on CNN design, we leverage ConvNext [Liu+22b] Blocks across our model, with depth-wise convolutions using 5×5 kernels. We empirically observe that ConvNext blocks achieve higher quality structural editions at a lower parameter count than vanilla U-Net blocks. To further help convergence and preserve details in the input images, we use residual connections [He+16; Dia+20; Rod+23a] in every convolutional block of the model. We use 1×1 convolutions on the skip connections and residual scaling [Kar+20a]. As in [Liu+22b], we use Layer Normalization [BKH16] and GELU non-linearities [HG16]. On the bottleneck of the model, we introduce an attention module [Woo+18] to help the model learn longer-range dependencies. To avoid checkerboard artifacts [ODO16], we use nearest neighbors interpolation for upsampling. Inspired by recent work on tileable material generation [Zho+22], we use *circular padding* throughout the model. We initialize its weights using *orthogonal initialization* [ACB19], which helps avoid exploding gradients.

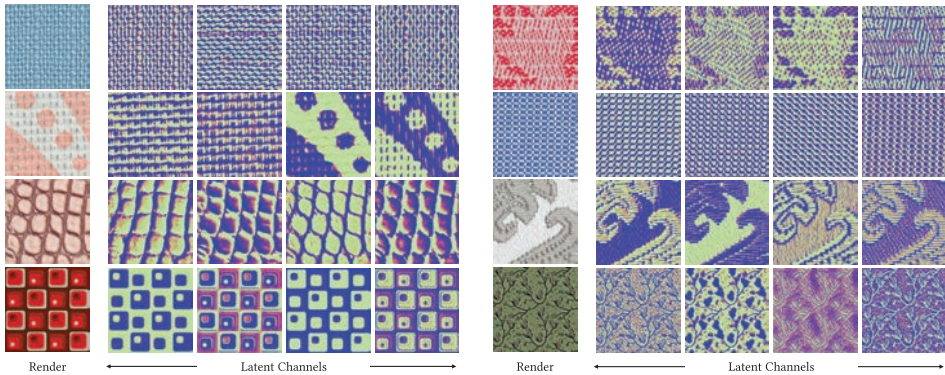
Renderer For the *renderer*, we build upon SIREN [Sit+20] MLPs, with additional modifications to enhance its performance for our problem. We use 1×1 convolutions instead of vanilla linear layers, to allow for end-to-end training using 2D images. Further, we introduce Layer Normalization [BKH16] before each sinusoidal non-linearity, which stabilizes training. Finally, inspired by [Meh+21], we use residual connections [He+16], to help preserve the information of the input vector across the decoder layers. Model weight initialization follows [Sit+20]. With sinusoidal activations, we observe significantly higher reconstruction quality and training dynamics than with ReLU [NH10] MLPs, which are common for BTF compression [Rai+19; Rai+20; Kuz+21]. Because the network is fully-convolutional, it can take as input feature vectors of any size. This is very convenient for our use cases when the input guidance image have a size different from the size of the original BTF used to train it.

Data Generation We generate synthetic ground truth BTF data using a proprietary CPU path-tracer based on the architectures of PBRT [PJH20],

Mitsuba [Nim+19] and Wave-front [LKA13]. The scene is set up with a single plane-mesh, and a directional light with unit irradiance rotating around the surface. We use a custom camera for skewed configurations to avoid post processing. We render a total of $81 \times 81 = 6561$ HDR images, changing the camera and lighting angles for each, following [SSK03]. The data are generated with 16 samples per BTF texel using a 10-core CPU, which takes approximately one hour for 512×512 resolution materials. For the rendered SVBRDFs, we use the material model in [Gar+23] (Chapter 2.C), without the transmittance and opacity maps. We use both procedurally and scanned SVBRDFs to generate our synthetic BTFs.

For acquired BTFs, we test on a variety of publicly available datasets, including *UBO 2003* [SSK03], *UBO 2014* [Rei+14], *Atrium*, *UTIA 2012* [Hai12], or *UTIA MAM 2014* [Fil+18]. Each of those has their own spatial and angular resolutions, we refer the reader to their individual specifications and to the Survey in [HS13] for additional details. We use *btf-extractor* to read the acquired BTF files. Our method works seamlessly with both measured and synthetically generated data without the need to change neither the neural module or the training procedure.

Figure 4.5: A selection of latent channels learned by NeuBTF for a variety of materials. We use a colorspace to help visualization. Without explicit supervision, the model internally learns semantically meaningful latent spaces. For instance, in the second example on the left, the two leftmost latent spaces encode geometry, while the other two encode the two distinct colors of the printed pattern over the yarns.



Optimization We use PyTorch [Pas+19] and Torchvision [MR10] for training. We train our models for 300 epochs, using a batch size of 80, and an input crop size of 128×128 pixels, as in [RG21; RG22; Rod+23a]. We set an initial learning rate of 0.0015, which we divide by 2 every 100 iterations using a step scheduler. We empirically observe training instabilities using Adam [KB15] on early iterations.

To mitigate this, we use RAdam [Liu+19a], which significantly stabilizes training. We also perform gradient norm clipping [Zha+20] (set to $\text{maxnorm} = 10$) and an $\epsilon = 10^{-6}$. We use a Dropout [Sri+14] rate of 0.1 on the autoencoder and a weight decay of 10^{-6} , to avoid overfitting. To further regularize the models, we leverage mixed precision training and automatic scaling [Mic+18]. This training process takes approximately 3 hours per material on a Nvidia RTX 3060 GPU. These training times are constant regardless of the material spatial resolution. For comparison, our model takes around four times longer to train than previous work [Kuz+21], because we need to optimize the autoencoder parameters and our loss function is multifaceted. Convergence is typically achieved on 100-150 iterations, longer training only helps resolve additional details. A subset of the training set is also leveraged as a validation set, which helps track the model performance on larger images. We use this validation dataset for early stopping regularization. During test time, we evaluate the model using *half precision*.

Loss Function We weight each part of the loss function as: $\lambda_{\text{style}} = 0.25$, $\lambda_{\text{freq}} = 1$, and $\lambda_c = 5$. For λ_{style} , we use the *AlexNet* variant of *LPIPS* [Zha+18], which allows for a lightweight yet powerful texture regularizer [RG22; Rod+23a]. This metric expects input on the $[-1, 1]$ range, so we perform a simple $2x - 1$ transformation to the model target and output, which lie on the $[0, 1]$ range before computing the loss.

Data Augmentation In order, we apply to both input and target images: random rescales in the range of $[0.25, 1.5]$ and random cropping. Then, only to the input image: random hue changes, random grayscale ($p = 0.2$), random blurs ($p = 0.2$, kernel size = 5) and random Gaussian noise ($p = 0.2$). We empirically observe that higher probabilities, or more complex policies like random erasing [Zho+20] were detrimental to the model performance for some materials. We use Kornia [Rib+20] for these augmentations.

Model Sizes Our *autoencoder* has 4 hidden layers in the encoder, with a *width factor* of 5. All other model details follow [RFB15]. The outputs of the autoencoder (e.g. the 2D latent space) are regularized with Layer Normalization [BKH16]. We use $D = 14$ latent dimensions for our neural materials, following [Kuz+21]. For the *decoder*, we use 3 hidden layers with 40 neurons each, achieving similar model sizes as previous work on neural BRDF representations [Kuz+21; Szt+21]. Following [Kuz+21], introduce a ReLU non linearity [NH10] to the final layer of the decoder to avoid negative reflectance values.

4.5 Evaluation

4.5.1 Qualitative Analysis

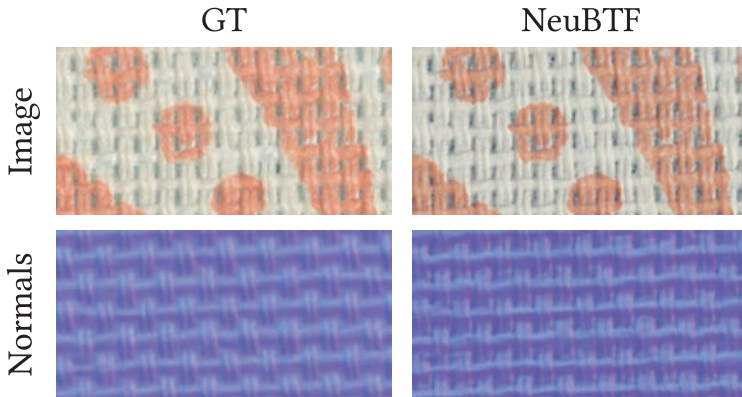
We evaluate NeuBTF on materials from different sources including acquired BTFs from [WGK14; Hai12], and rendered BTFs from procedurally generated and

scanned SVBRDFs. In Figure 4.3, we show examples of the results of NeuBTF for a variety of materials with highly complex structures and reflectance properties, like colored specular (first column) or anisotropy (last). We show some additional results in Figure 4.4 for materials of different datasets. As shown, our model achieves high quality reconstructions regardless on the type of data source.

In Figure 4.5, we show a colored visualization for a few channels of the latent neural texture T found for a variety of materials. Because the values for the neural texture are unbounded, to each channel $c \in T$, we standardize them to 0 mean and unit variance, and apply a $\text{sigmoid}(c) = \frac{1}{e^{1-c} + 1}$ non-linearity to make

the maps comparable. Without any explicit training, the models learn to separate distinct parts of the material. For example, the model finds distinct latent spaces for *warp* and *weft* yarns on woven fabrics, or separation between color and geometric patterns. This disentanglement provides clues on why the material propagation is possible, and suggests potential future research directions for fine-grained neural material edition.

Figure 4.6: Surface normals estimated using *Photometric Stereo* [Ike81] for ground truth materials and the materials reconstructed by NeuMat. As shown, the encoded materials preserve geometric information without explicit training.



Finally, we also evaluate whether the encoded materials preserve the geometric properties in their training datasets. We leverage *Photometric Stereo* [Ike81] to compute surface normals of encoded and ground truth materials. We provide results in Figure 4.6, where we show that the reconstructed surface geometry is accurate albeit noisier, and preserves the structure of the material.

4.5.2 Compression comparisons with previous work

In Table 4.1, we show the number of trainable parameters on the decoders of different neural BTF compression algorithms. As shown, our model is competitive with previous work in terms of trainable parameters. This is achieved as we use more complex loss functions than previous work, which help regularize the models, and because our sinusoidal MLP achieves higher quality reconstructions for natural signals than ReLU MLPs, as shown in [Sit+20]. NeuMIP [Kuz+21] uses smaller MLPs, however, they require an additional decoder for their *neural offset* module, which helps them encode parallax effects (See Figure 4.7), for which our model struggles. Our decoder has one order of magnitude fewer parameters than [Rai+19; Rai+20], however, the method in [Rai+20] provides the benefit of fast encoding of new materials, while ours requires a different model for each new material.

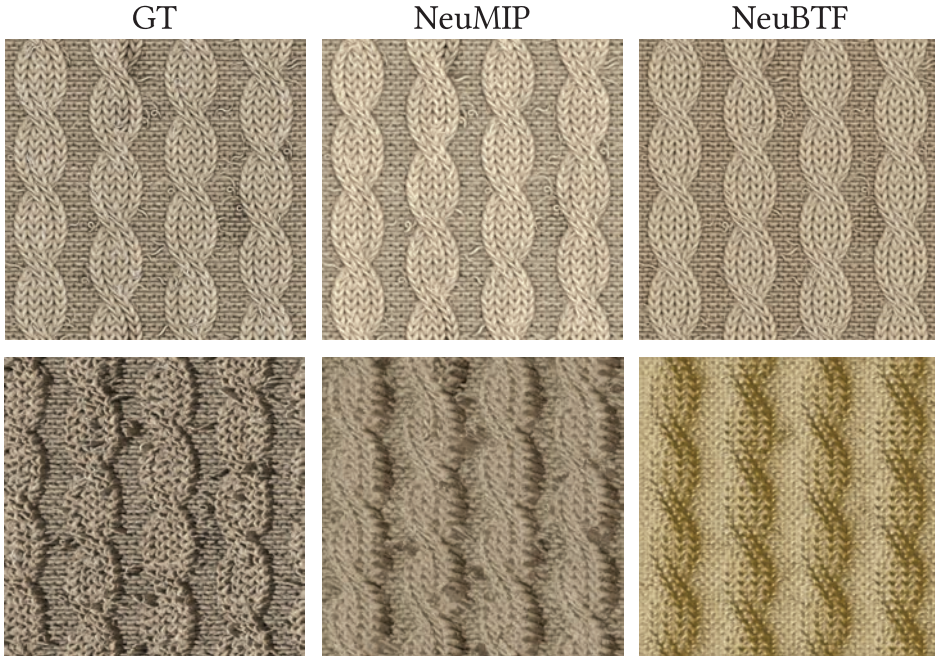
4.5.3 Limitations

As we show in Figure 4.7, our model struggles with materials with strong displacement. While our method provides accurate encodings on viewing angles close to the material surface, it cannot accurately encode grazing angles for such extreme cases. Displacement maps translate the geometric position of the points over the surface, breaking the underlying assumptions behind our neural texture. NeuMIP [Kuz+21] solves this issue by explicitly modelling parallax effects with a *neural offset* module. While we did not observe that such extension was needed for acquired BTF data, like the UBO2014 [Rei+14] dataset, introducing a similar module into our editable neural material framework is an interesting future research direction to increase its generality.

Table 4.1: Number of trainable parameters in the decoders and amount of latent texture channels for different neural BTF compression methods, measured using Torchinfo [Yep20]. Exact comparisons are challenging, as [Kuz+21] optimizes a multi-level texture pyramid and [Rai+20] learns a latent vector which can encode novel materials. For neither our method nor [Rai+19; Rai+20], we count the parameters in the encoders, as they are not needed for using the materials on rendering systems.

Method	NeuBTF (ours)	[Kuz+21]	[Rai+19]	[Rai+20]
Decoder Parameters	3011	3332	35725	38269
Texture Channels	14	14	14	38

Figure 4.7: A failure case of our method. Compared to NeuMIP [Kuz+21], which explicitly models parallax effects, our model struggles to accurately encode materials with strong displacements, as this synthetic cable knit from Substance3D. For this type of materials, NeuBTF accurately encodes orthogonal viewing angles (top row), however, it struggles at grazing angles (bottom row)



4.6 Applications

4.6.1 Reflectance Propagation and Tileable BTFs

Many material reflectance acquisition devices are limited in the surface dimensions they can digitize. This hinders their applicability to many real-world materials, which exhibit variations that cannot be captured at such small scales. Further, in many applications like SVBRDF acquisition, obtaining larger samples of the material improves realism and helps tileable texture synthesis. In this context, previous work on BTF reflectance compression inherits the surface area limitations of the capture devices used to generate their training data. Our method can easily be applied for reflectance propagation. We build upon the work of Rodriguez-Pardo and Garces [RG21] and leverage our *encoder* to propagate the neural texture optimized using a small portion of the material (e.g. a 1×1 cm capture) to a larger portion of the same material,

represented with a *guidance image* captured using a commodity device like a flatbed scanner. Because our model is trained using a large amount of lighting conditions, as in [RG21], the propagation is invariant to how the images are illuminated. We show results of such pipeline in Figure 4.3. For instance, on the last row, we show an anisotropic and specular Silver Jacquard fabric, for which we generated a BTF by rendering a 1×1 cm SVBRDF. This small crop cannot represent the complex pattern in the fabric, which we show on the *guidance image*, which covers a 10×10 cm area. Using our encoder, we propagate the neural material to this guidance image, generating a new, high-resolution, latent space which we can render, enabling realistic material representations with a reduced digitization cost. This propagated neural material has a 2000×2000 texels resolution, and it requires no re-training during test time.

Relatedly, our propagation framework can also be easily leveraged for generating tileable BTFs. Given any *guidance image* of the material, we can generate a tileable version of it, either using manual editions by artists or automatic algorithms [Mor+17; Li+20; Rod+19; RG22]. With this tileable input guidance, we can use our autoencoder A to propagate the neural texture T , effectively generating tileable BTFs, as we show in Figure 4.3. This propagation algorithm can leverage state-of-the-art algorithms for tileable texture synthesis without any modification of our material model or training framework. Tileable BTFs were not achievable with previous approximations and this simple pipeline has the potential of enabling novel applications of this type of material representation in rendering scenarios.

4.6.2 Structural Material Edition

Besides propagating BTF measurements to larger portions of the material, NeuBTF allows for generating novel materials using structural editions. Given a trained NeuBTF and a guidance image representing some particular target structure, we can propagate the neural texture to this guidance image, generating high-quality neural materials which preserve the structure of the guidance image and the reflectance properties of the trained neural material. This pipeline allows for easily generating multiple different neural materials without the need for retraining. As we show in Figure 4.8, this propagation method works for many types of input guidances, including vector black and white images, procedurally generated textures, or real photographs of materials and textures. We show results on acquired BTFs and from synthetic BTFs, rendered from scanned and manually generated SVBRDFs. As shown, our propagation frameworks provides high-quality material editions, even for very challenging cases, like the *circles pattern*.

4.6.3 Multi Resolution Neural Materials

Another useful application enabled by our method is the generation of materials at different resolutions. Unlike previous work [Kuz+21], which explicitly optimizes a pyramid of levels of detail during training, we can generate materials at any resolution at test time without introducing any additional complexities to our material representation. Because we train our models using random rescales as a data augmentation policy, they are equivariant to rescales of its input guidance images $G \downarrow: R(A(G)) \downarrow = R(A(G \downarrow))$. As such, we can generate any continuous resolution for a particular BTF by downsampling the guidance image to the target resolution and propagating its neural texture, as we illustrate in Figure 4.9. Note that this algorithm only guarantees accurate results for the rescaling ranges that we use during data augmentation.

4.7 Conclusions

We have presented a learning based representation for material reflectance which provides efficient encoding and powerful propagation capabilities. Our method introduces input conditioning into neural BTF representations. This allows for multiple applications which were not possible with previous neural models, including BTF extrapolation, tiling, and novel material synthesis through structure propagation. Our method builds upon recent work on neural fields, network design and data augmentation, showing competitive compression capabilities with previous work on neural BTF representation. Through multiple analyses, we have shown the capabilities of our method on a variety of materials with different reflectance properties, including anisotropy or specular, as well as effectively handling either synthetic or acquired BTFs.

Our method can be extended in several ways. The most immediate extension is to allow for materials with strong parallax effects due to displacement mapping or curvature, as in [Kuz+21; Kuz+22]. Our representation is limited to opaque materials. Extending them to handle translucent or holed surfaces would increase their realism in materials like thin fabrics or meshes. Further, our method could be extended to allow for hyperspectral BTF data [RSK10], but captured data is scarce. Besides, recent work on neural BRDF representations [Szt+21; Rai+22; Fan+22] and generative models [Mül+19] suggests a promising research direction: Learning to sample from neural BTFs, using invertible neural networks. While these may introduce challenging complexities to the models, they could provide efficient representations for Monte Carlo rendering using importance sampling. Further, building upon recent work on SVBRDF capture [Rod+23a; Zho+22], BRDF sampling [NJR15] and BTF compression

[Rai+20], it could be possible to learn a prior over neural BTFs with a generative model. This should help in capturing more efficiently the data needed for generating these assets, as well as generating new materials and interpolating between them. Finally, editing semantic and reflectance properties in neural fields is an active area of research [WTX22; Liu+21; Wan+22a; Ye+22; Haq+23]. While our method introduces structural edition into neural BTF representations, it is not capable of editing particular semantic properties, such as albedo or specular. Extending our edition capabilities to more fine-grained parameters is an interesting research avenue. We hope our method inspires future research on neural material representations.

Figure 4.8: Examples of structural editions allowed by our method on a variety of materials. On the leftmost column, we show *guidance images*, which represent structures into which we transfer the neural BTF measurements illustrated on the top row. As shown, our method can effectively propagate BTF measurements into many material and structure types, using as guidances either synthetic or real images. The first three materials (*leather11*, *carpet07*, *fabric01*) are taken from *UBO 2014* [Rei+14], the *linen* material is rendered from a captured SVBRDF, while the *ceramic* material is rendered from an artistic material taken from *Substance3D*. We show renders generated using $\theta_v = \theta_l = 0$.

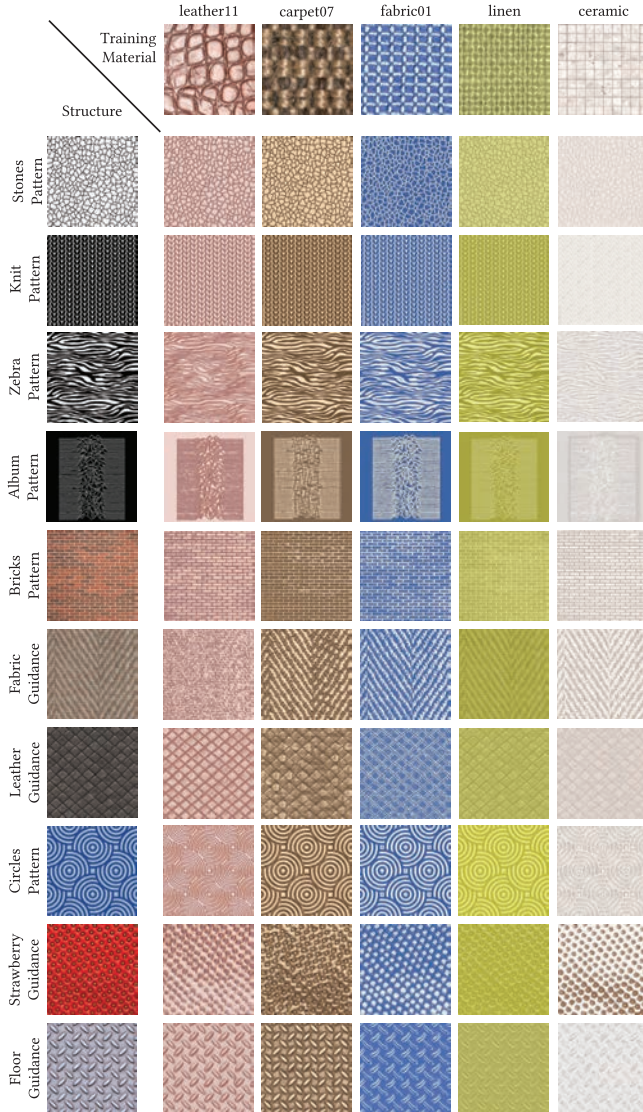
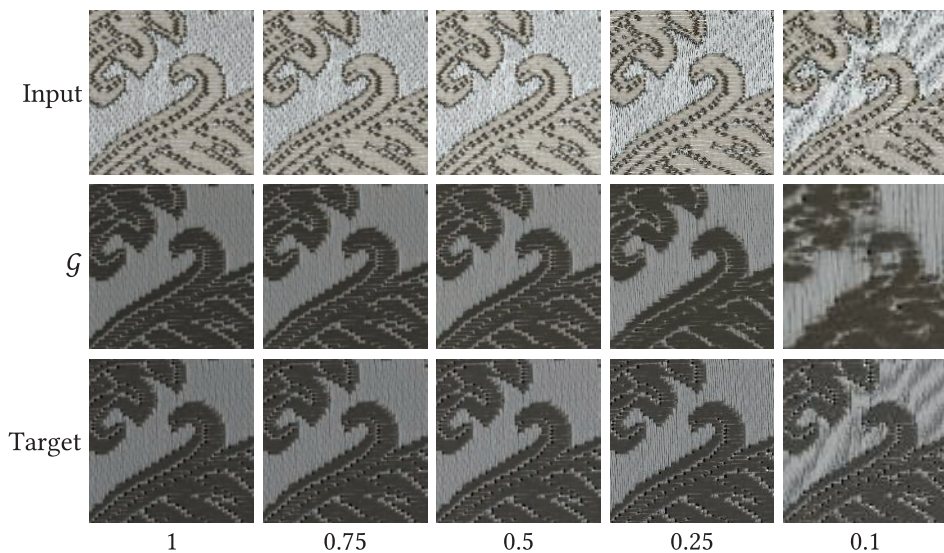


Figure 4.9: Our method naturally enables for the generation of different resolutions for the neural materials. We show the input to the autoencoder (top row), the rendered material at $\theta_c = 75, \theta_l = 60, \phi_c = \phi_l = 0$ (middle row), and the ground truth image at those positions (bottom), at different resolutions (columns). We achieve this by downsampling the guidance image G fed into the autoencoder module, which returns an accurately down-sampled latent space, thanks to our data augmentation policy applied during training. The rightmost column lies beyond the ranges in which we train the model, however, the results are still somewhat plausible.



4.A Model Implementation Details

Figure 4.10: A full diagram of our U-Net autoencoder A. We use a CBA77777M [Woo+18] module on the bottleneck, Layer Normalization [BKH16], and residual connections in every convolutional block. Inspired by ConvNext Blocks [Liu+22b], we use 5×5 depthwise convolutions followed by 1×1 linear layers, with GeLU non-linearities [HG16]. We use 14 latent dimensions for our neural materials, following [Kuz+21]. We use *circular padding* throughout the model and initialize its weights using *orthogonal initialization* [ACB19], which helps avoid exploding gradients.

In red, we show the input/output dimensions of each layer; in orange, we show attention modules; in blue, convolutional blocks and layers; in green, upsampling and concatenating operations; in yellow, normalization layers; and in purple, non-linearities.

Autoencoder $\mathcal{A}(I) = \mathcal{T}_I$

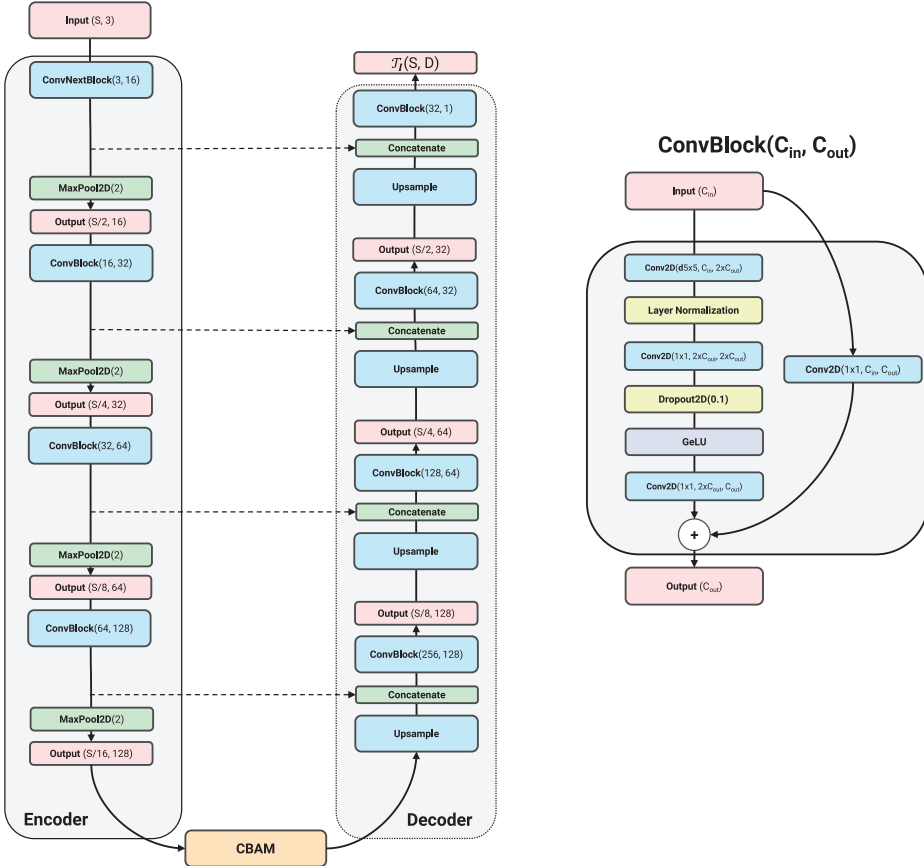
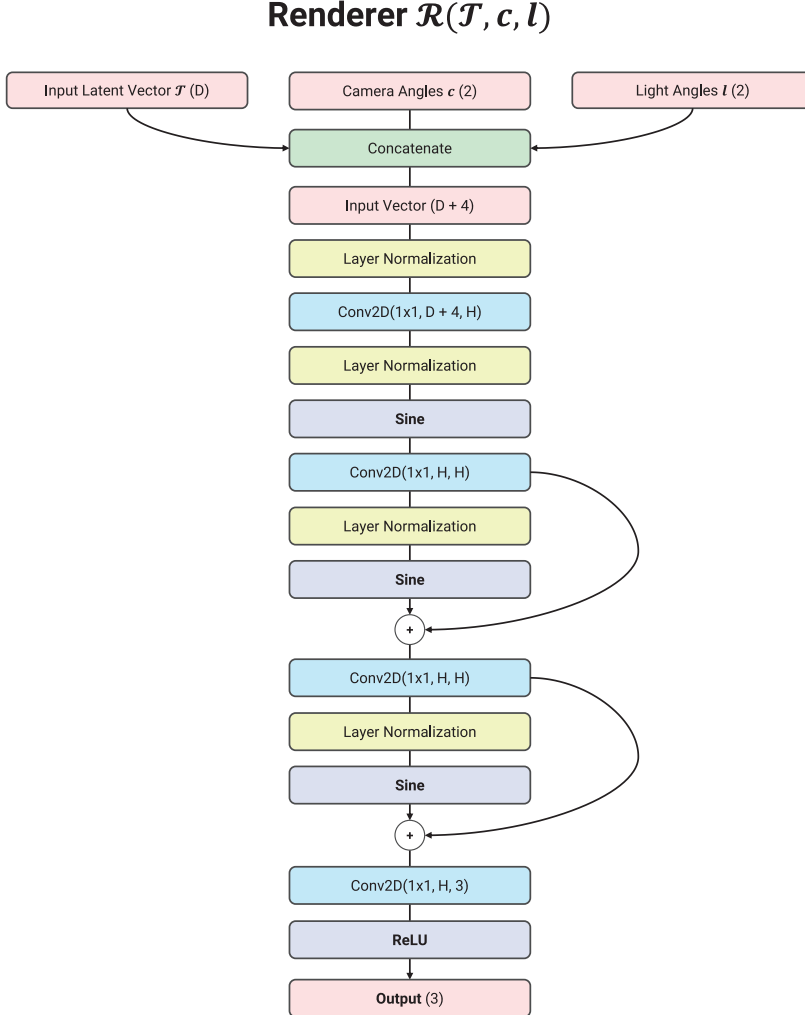
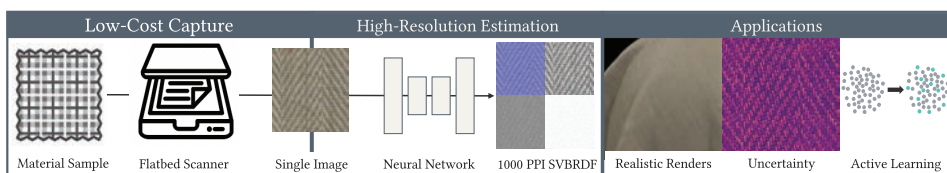


Figure 4.11: A full diagram of our sinusoidal renderer R . We build upon a SIREN architecture [Sit+20], which takes as input the latent vector, queried from the input texture, as well as target camera and light angles to re-render. We introduce Layer Normalization after each linear layer, which we implement using 1×1 convolutions to allow for 2-D inputs during training. The output of the model is clamped to avoid negative output radiance values, using a simple ReLU non-linearity [NH10], following [Kuz+21].



CHAPTER 5. UNCERTAINTY-AWARE NEURAL NETWORKS FOR HIGH RESOLUTION DIGITIZATION

Figure 5.1: From a single image captured with a flatbed scanner *UMat* estimates high-resolution SVBRDFs. Leveraging an attention-guided generative model, the estimations are sharp and artifact-free, and can be used in photorealistic rendering. We introduce a novel uncertainty quantification algorithm for material digitization, which correlates with error at test time and can be used for dataset creation.



In this chapter, we introduce a learning-based method to recover normals, specularity, and roughness from a single diffuse image of a material, using microgeometry appearance as our primary cue. Previous methods that work on single images tend to produce over-smooth outputs with artifacts, operate at limited resolution, or train one model per class with little room for generalization. In contrast, in this work, we propose a novel capture approach that leverages a generative network with attention and a U-Net discriminator, which shows outstanding performance integrating global information at reduced computational complexity. We showcase the performance of our method with a real dataset of digitized textile materials and show that a commodity flatbed scanner can produce the type of diffuse illumination required as input to our method. Additionally, because the problem might be ill-posed (more than a single diffuse image might be needed to disambiguate the specular reflections) or because the training dataset is not representative enough of the real distribution, we propose a novel framework to quantify the model's confidence about its predictions at test time. Our method is the first one to deal with the problem of modeling uncertainty in material digitization, increasing the trustworthiness of the process and enabling more intelligent strategies for dataset creation, as we demonstrate with an active learning experiment. A version of the model presented in this chapter is part of [Textura.ai](#). The contributions in this chapter led to the following publication¹:

¹ Additional publication details and results are included on [the project website](#)

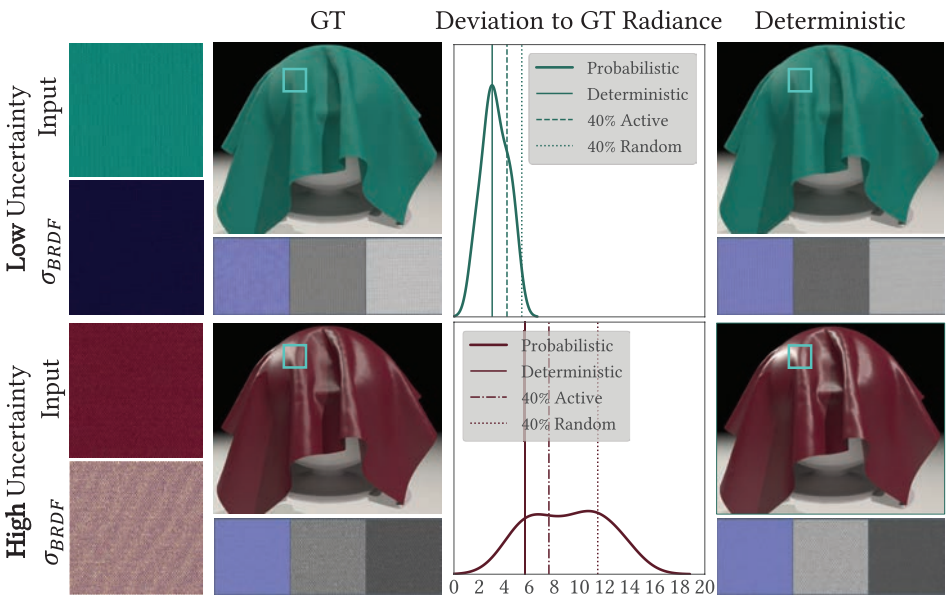


“UMat: Uncertainty-Aware Single Image High Resolution Material Capture”

Carlos Rodriguez-Pardo, Henar Dominguez, David Pascual, Elena Garces

Proc. of Computer Vision and Pattern Recognition (CVPR) (2023)

Figure 5.2: Our method digitizes a material taking as input a single scanned image. Further, it returns a pixel-wise metric of uncertainty σ_{BRDF} computed at test time through probabilistic sampling, proven useful for active learning. In the plot we compare the average deviations of the radiance of different renders in the blue crop w.r.t the ground truth (GT) of: 1) the distribution of the probabilistic samples of a model trained with 100% of the data; 2) the deterministic output of that model; 3) the output of a model trained using 40% of the training dataset, sampled by active learning guided by σ_{BRDF} and; 4) a model trained using 40% of the training dataset, randomly sampled. The material at the bottom, for which the model shows a higher uncertainty, generates more varied renders and differs most from the ground truth.



5.1 Introduction

Virtual design, online marketplaces, product lifecycle workflows, AR/VR, video-games, . . . , all require lifelike digital representations of real-world materials (*i.e.*, digital twins). Acquiring these digital copies is typically a cumbersome and slow process that requires expensive machines and several manual steps, creating roadblocks for scalability, repeatability, and consistency. Among the many industries requiring digital twins of materials, the fashion industry is in a critical position; facing the demand to digitize hundreds of samples of textiles in short periods, which cannot be achieved with the current technology.

In this context, casual capture systems for optical digitization provide a promising path for scalability. These systems leverage handheld devices (such as smartphones), one or more different illuminations, and learning-based priors to estimate the material's diffuse and specular reflection lobes. However, existing approaches present several drawbacks that make them unsuitable for practical digitization workflows. Generative solutions [Guo+20b; VPS21] typically produce artifacts and are generally not tileable. Despite recent attempts to improve tileability and controllability [Zho+22], these solutions are slow to train and evaluate (requiring online optimization iterations), are limited in resolution, and present challenges for generalization (requiring one model per material class). Further, the fact that these methods build on perceptual losses ~~shot pixel losses~~ to compare the input photo with the generated material entails extra difficulties when it comes to guaranteeing the repeatability and consistency required for building a digital inventory (*i.e.*, color swatches, prints, or other variations). On the other hand, methods that build on differentiable node graphs [Hen+21] overcome the tileability and resolution limitations, yet, they share the problems derived from using perceptual losses and category-specific training.

In this chapter, we present *UMat*, a practical, scalable, and reliable approach to digitizing the optical appearance of textile material samples using SVBRDFs. Commonly used SVBRDFs typically contain two reflection terms: a diffuse term, parameterized by an albedo image, and a specular one, parameterized by normals, specularity, and roughness. Prior work typically estimates both

Figure 5.3: Scanner images vs fitted albedos.



components, which becomes a very challenging problem, obtaining over-smooth outputs, and being prone to artifacts [Des+18; Guo+21; ZK21]. Instead, in this chapter, we demonstrate that it is possible to provide accurate digitizations of materials leveraging as input a single diffuse image that acts as albedo and estimating the specular components using a neural network.

Our key observation is to realize that most of the appearance variability of textile materials is due to its microgeometry and that a commodity flatbed scanner can approximate the type of diffuse illumination that we require for the majority of textile materials (see Figure 5.3).

Nevertheless, single-image material estimation is still an ill-posed problem in our setting, as reflectance properties may not be directly observable from a single diffuse image. To account for these non-directly observable properties, we propose a novel way to measure the model's confidence about its prediction at test time. Leveraging Monte Carlo (MC) Dropout [GG16], we propose an uncertainty metric computed as the variance of sampling and evaluating multiple estimations for a single input in a render space. We show that this confidence directly correlates with the accuracy of the digitization, which helps identify ambiguous inputs, out-of-distribution samples, or under-represented classes. Besides increasing the trustworthiness of the capture process, our confidence quantification enables smarter strategies for dataset creation, as we demonstrate with an active learning experiment.

We pose the estimation as an Image-to-Image Translation problem (I2IT) that directly regresses roughness, specular, and normals, from a single input image. Under the hood, our novel residual architecture has a single encoder enhanced with lightweight attention modules [Wan+20b; MR21] for improving global consistency and reducing artifacts, specialized decoders for each target reflectance map, and a U-Net discriminator [SSK20], which enhances generalization.

In summary, we present the following contributions:

- A novel material capture system which leverages the diffuse illumination provided by flatbed scanners for high-resolution, scalable, and reliable digitizations.
- An attention-enhanced GAN model and training procedure designed for maximizing accuracy and sharpness, and removing undesired artifacts.
- A generic uncertainty quantification framework for material capture algorithms which correlates with prediction error on a render space.
- An exhaustive evaluation method for measuring the accuracy and quality of the model estimations.

5.2 Related Work

5.2.1 Lightweight Material Capture

Single Image SVBRDF Capture Capturing accurate SVBRDFs from a single image is a challenging problem that requires predicting the photometric response of a material given only a sample of it. The most common approximation is to use a flash-lit front planar image captured with a smartphone. Extending neural style transfer [GEB15b] to material capture, Aittala *et al.* [AAL16] leverage pre-trained CNNs and texture priors for smartphone material acquisition. Relatedly, Henzler *et al.* [Hen+21] use style losses for training generative models for BRDF synthesis. These approaches are optimized for synthesis, which allow for seamlessly tileable outputs, and do not require supervised training. However, they work best for stochastic materials, limiting their scope.

Leveraging datasets of labeled materials, different methods have trained autoencoders for SVBRDF capture. Li *et al.* [Li+17a] reconstruct spatially-varying albedo and normals using a U-Net [RFB15], and homogeneous specular albedo and roughness using a CNN regressor. This work was extended through Self-Augmented CNNs in [Ye+18]. By leveraging Conditional Random Fields (CRFs), a material classifier and one decoder per map, Li *et al.* [LSC18] reconstruct spatially-varying albedo, roughness, and normals. Deschaintre *et al.* [Des+18] propose a modified U-Net, synthetic datasets, and a render loss. Cascaded models [Li+18; SC20]; and deep latent spaces optimized using inverse rendering [Gao+19] have also shown success for this problem. Recently, Generative Adversarial Networks (GANs) have shown improved capabilities compared to more naive losses. These require a discriminator, which can be trained on renders [Wen+22], SVBRDF maps [Guo+21; VPS21], or both [ZK21].

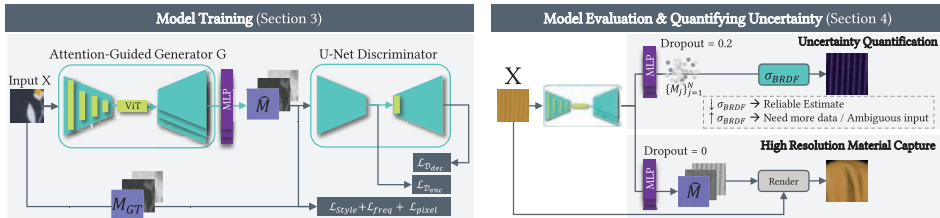
Our approach differs from previous work in several factors. Importantly, we use flatbed scanners instead of smartphones for material capture. While they limit the materials which can be captured, they provide adequate illumination for easier digitizations, and a higher level of resolution and detail. We hypothesize that material specularity can be estimated accurately by leveraging its microgeometry. From this assumption, we build a GAN which, in contrast with previous work, leverages state-of-the-art attention mechanisms and discriminator design for obtaining a more holistic understanding of its inputs. Further, we train exclusively on real data, and propose a more comprehensive evaluation.

Multiple Image SVBRDF Capture A different corpus of work relies on multiple images of the material. By capturing more evidence of the material photometric response, they provide more accurate SVBRDFs. This hinders scalability, as they require a larger capture and calibration effort. The simplest approach is to require two samples, a diffuse and a flash-lit image [AWL15; Bos+20]. More

flexible alternatives allow for more samples, combined with a learned prior [Guo+20b; Des+19]; or Monte Carlo rendering [Lua+21]. A different approach is to capture the material at a high resolution, and transfer those details to larger samples of the same material [DDB20; RG21]. Finally, videos for material acquisition have also shown to be promising capture systems [Ye+21].

Procedural Graphs Procedural graphs have also been used for material generation. Instead of relying on material priors, these approaches work by optimizing a material graph through a differentiable pipeline. These provide interesting capabilities, such as easy edition or tiling, but are limited by the expressiveness of the procedural model. These have been explored for general SVBRDF estimation [Shi+20; Hu+22c; Guo+20a], or for high-quality woven fabric digitizations [Jin+22].

Figure 5.4: Overview of *UMat*. We propose an attention-guided generator trained with style, frequency, pixel-wise, and adversarial losses. In green, we show the components that include any form of attention mechanism. The supplementary material contains the detailed architectures. On the right, we show two applications of our method: First, thanks to our test-time uncertainty quantification, we can provide a measure of the reliability of the estimation. Second, the maps that *UMat* produces can be used by any render engine.



5.2.2 Uncertainty Quantification in Deep Learning

Measuring the confidence of deep learning models is an active research area [KG17] with multiple applications, including safety-critical problems, like self-driving [Hua+19] or medicine [Kur+22]; and active dataset creation [Lei+21; Sol+21]. In computer vision, uncertainty quantification has focused on image classification [SKK18; Lei+21], segmentation [Kry+21], and depth regression [Hu+22a; Ami+20]. To overcome the computational intractability of Bayesian Neural Networks, different approximations have been proposed, including MC Dropout [GG16], Deep Ensembles [LPB17] or Variational Inference [MNG17]. Orthogonal alternatives exist, including Evidential Deep Learning [Ami+20; SKK18] or frequentist approaches like Constrained Ordinal Regression [Hu+22a]. We refer the reader to recent surveys [Jos+22; Gaw+21] for more comprehensive reviews.

Quantifying uncertainties allows communicating the end user that the model predictions may be inaccurate, suggesting alternative pathways; as well as cheaper dataset creation through *active learning*. Bayesian material parameter estimation has been proposed for procedural frameworks [Guo+20a], but, to the best of our knowledge, it has not been explored for SVBRDF estimation. We aim to propose an efficient uncertainty quantification framework for deep SVBRDF capture methods which accounts for material perception and is agnostic to the material model.

5.3 Method

Our method starts from an input image X of a material taken under diffuse lighting and outputs the parameters of the specular lobe of the SVBRDF, *i.e.*, $\mathbf{M} = \{\mathcal{M}_i\}_{i=1}^3$ corresponding to the material roughness, specularity, and normals. We illustrate this process in Figure 5.4.

Following previous work [KG13; Mun+22], we use the physically-based material model from Disney [BS12], which aggregates a diffuse term with an isotropic, microfacet specular GGX lobe $s(\mathbf{M})$ [Wal+07], such that, $f_{l,v}(\mathbf{M}, X) = \frac{X}{\pi} + S_{l,v}(\mathbf{M}) \in \mathbb{R}^{x \times y}$ is the shading model for a light l and view position v . We formulate this estimation as an Image-to-Image Translation problem (I2IT). We train a U-Net [RFB15] generator $G(X) = \hat{\mathbf{M}}$ within a GAN framework, extending the adversarial loss with pixel-wise, style, and frequency losses. We design the generator to maximize accuracy, sharpness, and robustness. To do so, we train a single encoder, enhanced with self-attention layers and a transformer, and use one decoder per map. Sections 5.3.1, 5.3.2, and 5.3.3 present the network design, the loss and data augmentation choices, respectively. Implementation details are provided on the supplementary material (Section 5.A).

Using as input a single image taken under diffuse lighting presents extra challenges when estimating the SVBRDF; we lack the extra cues provided by more complex illumination patterns (e.g. flash lighting). Therefore, in Section 5.4, we explain how to compensate for this potential ambiguity by introducing an uncertainty metric that can be computed at test time. Section 5.5 presents the evaluation, which includes the description of our dataset and metrics (Sections 5.5.1 and 5.5.2), an ablation study that validates the design (5.6.1), qualitative and quantitative results (5.6.2 and 5.6.3), an application of our uncertainty metric in an active learning setting (5.6.4), and comparisons with previous work (5.6.5).

5.3.1 Network Design

We use a U-Net [RFB15] with residual connections [He+16; Dia+20; DLG21] in all of our convolutional blocks, an individual decoder per map [Gar+22; RG22;

DLG21; ZK21], and group normalization [WH18]. To each decoder, we add a pixel-wise dropout-regularized MLP [Sri+14], aimed at increasing the accuracy of the predictions and allowing us to measure uncertainty at test-time.

While this multi-decoder residual U-Net is relatively accurate, it is limited by its receptive field, as is common on fully-convolutional architectures. Previous work [Des+18] proposed the use of a *global track* for fusing spatially distant information. We instead draw inspiration from recent advances in attention and diffusion models [Sah+22; HJA20; Rom+22], and add a self-attention module with linear complexity [Wan+20b] to the output of every convolutional block in the encoder. Finally, we add a lightweight MobileViT [MR21] to the bottleneck to provide the model with a global understanding of its input. By performing the most complex computations at the encoder, we provide the specialized decoders with dense inputs which are computed only once.

5.3.2 Loss Function

Our loss function is comprised of four terms: pixel-wise losses, an adversarial loss, a style loss, and a frequency loss:

$$\mathcal{L}_G = \sum_i \lambda_i \mathcal{L}_{pixel_i} + \lambda_{adv} \mathcal{L}_{adv} + \lambda_{style} \mathcal{L}_{style} + \lambda_{freq} \mathcal{L}_{freq} \quad (5.1)$$

$\mathcal{L}_{pixel}$ is the L_1 norm weighted per map, λ_i . L_1 produces sharper results than higher-order alternatives, such as L_2 . We introduce an adversarial loss to handle the intrinsic ambiguity of ill-posed problems [Tex+20b; Iso+17; Gar+22; VPS21]. In our case, the choice of the discriminator is a critical design decision.

Recent work [SSK20] proposed U-Net architectures for discriminators, which allows for better learn both low and high-level features, and introduces further regularization. These result in more conservative albeit less diverse generations [Han+22a]. We use a U-Net discriminator [SSK20] with attention [Woo+18], which outputs two estimations: a scalar output D_{enc} , provided by its encoder, and a 2D estimation D_{dec} provided by its decoder. D_{enc} provides a global estimate of the quality of the stack M , while D_{dec} gives pi estimations. As in [SSK20], we add a *regularization* term $\mathcal{L}_{D_{dec}}^{cons}$, and leverage *cut-mix* as for discriminator data-augmentation. Our discriminator and adversarial losses are:

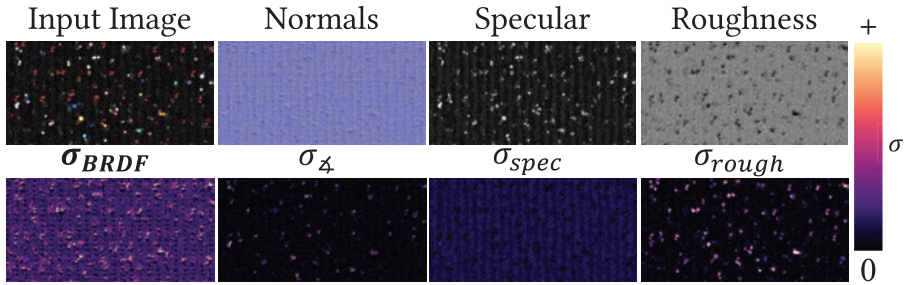
$$\mathcal{L}_D = \mathcal{L}_{D_{enc}} + \mathcal{L}_{D_{dec}} + \lambda_{cons} \mathcal{L}_{D_{dec}}^{cons} \quad (5.2)$$

$$\mathcal{L}_{adv} = \log\left(\mathcal{D}_{enc}\left(G(X)\right)\right) + \log\left(\mathcal{D}_{dec}\left(G(X)\right)\right) \quad (5.2)$$

where the implementation of $\mathcal{L}_{D_{enc}}$ and $\mathcal{L}_{D_{dec}}$ follows [SSK20].

We further add two losses to improve the accuracy and sharpness of the results: a frequency loss L_{freq} and a style loss L_{style} . L_{freq} is estimated by averaging the *focal frequency loss* [Jia+21] computed over each individual channel of M . This is designed to help GANs preserve high-frequency details. Further, as shown in prior work, working in the frequency domain is beneficial when handling textures [AWL13; Mar+20]. Our style loss L_{style} is inspired by the success of neural losses when dealing with textures [GEB15b; GEB15a]. However, off-the-shelf metrics which are designed for 3-channel images, are not immediately usable in SVBRDF. While it is possible to compute them for each map separately [RG22], this does not necessarily preserve inter-map consistency. We follow recent work [CHB21] and use LPIPS [Zha+18] taking as input a 3-channel image created by randomly sampling three channels from the set of five available channels of M .

Figure 5.5: Top: input image of a *rib* material with metallic sequins. Bottom: σ_{BRDF} and per-map uncertainties.



5.3.3 Data Augmentation

We follow two strategies for data augmentation. First, we perform patch-based training and affine transforms with randomly cropped patches [RG21; Tex+20b; VPS21]. We also apply random rescales for generalization at lower resolutions, and rotations to account for possible misalignments that may arise when capturing the samples. Second, we apply several image transformations to increase model robustness: random intensity changes in HSV space to the inputs, gaussian noise and blurs, and random erasing [Zho+20] for regularization.

5.4 Uncertainty Quantification

Material acquisition from a single diffuse image is potentially an ill-posed problem, since it assumes that the microgeometry is a sufficient cue to predict the material appearance. While purely learned priors have been proven to work well for inverse problems of single images object shape estimation [Lic+21; Lua+21; Hwa+22; Mun+22; Li+18], they are typically combined

with render losses that guarantee consistency in the reconstruction. We lack the necessary input to include this kind of supervision, therefore, we propose an alternative approach aimed at quantifying the confidence of the prediction.

This is valuable for several purposes. First, it provides a way of communicating possible inaccuracies to users of these systems when it is not possible to have access to the ground truth reflectance values; and, importantly, it enables efficient dataset creation through active learning, as we show in Section 5.6.4.

We propose an uncertainty quantification mechanism that is applied to individual per-map estimations, and also globally in a render space.

It is possible to measure uncertainty, among other methods, through deep ensembles [LPB17], evidential learning [SKK18; BYK21; Wan+22b], or ordinal regression [Hu+22a]. However, they are costly to train and evaluate, and would imply major changes in our method. Instead, we follow a probabilistic approach called MC Dropout [GG16], with which, for a particular input, we sample a set of predictions by adding randomness to the forward pass of our model. This process has no impact in the regular deterministic evaluation and implies no changes in our model architecture. Specifically, while measuring uncertainty, we randomly deactivate 20% of the neurons of the MLP of our decoders, obtaining a set of outputs $\mathcal{U} = \{\hat{\mathcal{M}}_j \sim G(\mathbf{x})\}_{j=1}^N$ that we use to compute several metrics.

First, we compute the pixel-wise standard deviation for each individual map separately, obtaining σ_n , σ_{spec} , and σ_{rough} for the normals, specular, and roughness maps, respectively. Then, inspired by perceptual metrics for computing BRDF differences [Lav+21], we define our novel perceptually-aware uncertainty metric σ_{BRDF} as follows:

$$\sigma_{\text{BRDF}} = \frac{1}{|xy|} \sum_{xy} \log \left(\frac{1}{|S|} \sqrt{\sum_{(l,v) \in S} \sqrt[3]{\sigma_{l,v}^2 \left(\left\{ f_{l,v}(\mathbf{u}_j, K) \cos(\theta_l) \right\}_{j=1}^N \right)}} \right) \quad (5.4)$$

where K is a 2D grayscale image of a constant neutral grey value, $\sigma_{l,v} \in \mathbb{R}^{x \times y}$ is the pixel-wise standard deviation of the renders obtained for the set of sampled maps $\mathbf{U}$, and S is the fixed set of 50 views optimized in [NJR15] for efficient BRDF capture. The log spatially-varying result is integrated across the spatial dimensions xy to obtain a single value as output. This equation introduces a perceptual component to our uncertainty metric in two forms: first, by applying cosine weighting to the light position to compensate for light attenuation at grazing angles, and second, by taking the cubic root of these differences to attenuate peak reflectances. Figure 5.5 showcases an

example of the predicted uncertainties for a rib material with sequins. While the uncertainty is low at the yarns because it is a common material in our dataset, it appears high in the sequins since that effect had not been observed during training.

5.5 Experimental Results

5.5.1 Dataset

We gathered a novel dataset for training and testing our method. It comprises 2000 textile materials of a variety of families and microstructures that we divided into 14 families: crepe, jacquard, pile, plain, satin, and twill for *wovens*; fleece, french terry, interlock, jersey, milano, pique, and rib for *knits*; and, finally, *leathers*. Each family differs in its construction pattern

(i.e., its microstructure), which directly impacts its optical appearance. In the supplementary material, we show a more detailed analysis of the dataset. For each material, we have an image scanned with the flatbed scanner EPSON V850 whose lighting configuration is close to diffuse (as we show in Figure 5.3), and its corresponding ground truth specular maps of the SVBRDF (normals, specular, and roughness). To obtain these maps, first, we digitized the material with an optical gonioreflectometer and then, propagated the maps to the scan using map propagation techniques [RG21]. All our images and maps have a resolution of 1000 PPis, allowing us to leverage the full semantics of the microstructure for inference. We split our dataset into 90-10 for train and test, making sure that every family is equally represented in both splits.

5.5.2 Metrics

We quantify individual *per-map accuracy*, *rendered perceptual accuracy*, and *artifacts*. **Per-map accuracy** is computed differently depending on the semantics of the map: the roughness and specular maps are evaluated using Mean Absolute Error (L_1), and the normals are evaluated using the angular distance in vector space ($L<$). Further, to account for the possibility that the model always returns an accurate average value, resulting in relatively low L_1 , we also measure Pearson correlation ρ .

We evaluate **rendered perceptual accuracy** LBRDF between the ground truth stack $\mathbf{M}_{GT}$ and the estimation $\hat{\mathbf{M}}$ following existing metrics [Lav+21],

$$\mathcal{L}_{\text{BRDF}} = \frac{1}{|xy|} \sum_{xy} \sqrt{\frac{1}{|S|} \sum_{(l,v) \in S} \sqrt[3]{\cos^2(\theta_l) \left(f_{l,v}(\mathbf{M}_{GT}, K) - f_{l,v}(\hat{\mathbf{M}}, K) \right)^2}} \quad (5.5)$$

where the terms are the same as in Equation 5.4.

Finally, we found that some architectural improvements in the neural network introduce artifacts in the specular and roughness maps that were not present in the input image, as illustrated in Figure 5.6. Thus, we provide an **artifacts detection** metric to quantify them. We start by defining a metric of homogeneity for an input image $H(I)$,

$$\mathcal{H}(I) = \frac{1}{|xy|} \sum_{xy} \frac{1}{|d|} \sum_{d=\{\uparrow, \downarrow, \leftarrow, \rightarrow\}} \left\| F_{\text{Box}}(I) - F_{\text{Box}}(I^d) \right\|_1 \quad (5.6)$$

where I^d is the image shifted up, down, left, and right by a number of pixels equal to the kernel size of the box filter. Then, we define three metrics that we

compute per map: $e_2(\mathcal{M}) = \frac{\mathcal{H}(\mathcal{M})}{\mathcal{H}(X)}$, and $e_3(M) = MI(X, M)^{-1}$. X is the original input

image, and MI is the Mutual Information [Rus+04], which helps discern artifacts that appear in a single map from semantic patterns (e.g., plaids, prints). Each map is labeled as having artifacts if the majority of metrics exceed their corresponding thresholds, $t_m(M)$. If any of the maps contain artifacts, the entire stack $\mathbf{M}$ is classified as having artifacts.

5.6 Evaluation

5.6.1 Ablation Study

In Figure 5.6 and Table 5.1, we present an ablation study to validate our model design. From a baseline U-Net [RFB15] trained with a pixel-wise loss and no data augmentation other than rescales, we add different components to the model to improve its generalization. First, we observe that using a Patch-GAN [Iso+17; VPS21] discriminator provides a significant increase in accuracy. However, using a similarly-sized U-Net discriminator [SSK20], we achieve better results, particularly when using discriminator regularization. Further, $\mathcal{L}_{\text{style}}$ and $\mathcal{L}_{\text{freq}}$ yield significant improvements, most notably in the normal map. To the baseline U-Net trained with the full loss, adding residual connections and one decoder per map increases accuracy. However, this setup tends to produce artifacts as it struggles to integrate global information. By adding self-attention to the encoder, we remove the artifacts; and with the MobileViT [MR21], we achieve higher quality results. Our final model contains full data augmentation, which provides small gains on generalization.

5.6.2 Qualitative Analysis

We aim to understand which features are exploited by our model for making its predictions. In Figure 5.7, we show the embeddings of our generator using UMAP. It seems that our model learns to separate between material *families*

(e.g. *leathers* from *wovens*). Interestingly, the *thickness* of the material is also a relevant parameter. The model is exploiting these semantic patterns without explicit supervision, providing evidence that material microgeometry plays an important role in its optical appearance.

Figure 5.6: Qualitative results of some configurations of our ablation study. In **red**, we show that the baseline generator architecture trained on the full loss introduces artifacts, which are removed using **attention** on the encoder.

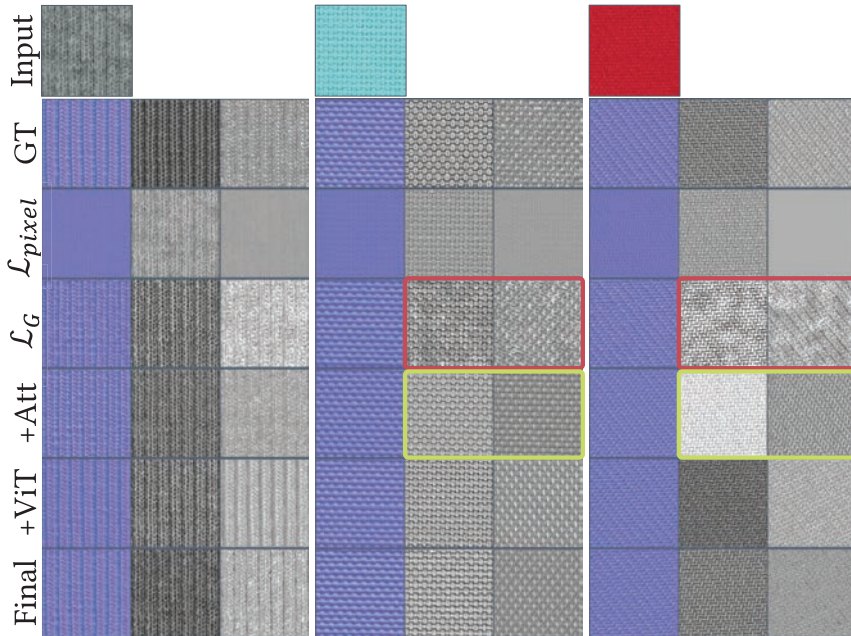


Figure 5.7: Embeddings of the transformer of our generator for data in the training set, reduced using UMAP [MHM18].

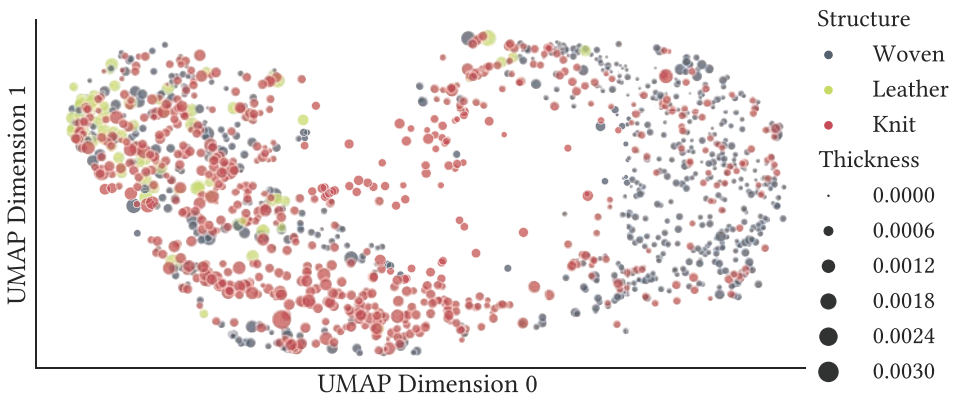


Table 5.1: Results of our ablation study, across a variety of metrics. *Art.* refers to our artifact detection metric. We use a color code to highlight **best** and **worst** cases.

	Configuration	$\rho^S \uparrow$	$\rho^R \uparrow$	$L_{\downarrow}$	$\mathcal{L}_q^S \downarrow$	$\mathcal{L}_q^R \downarrow$	$L_{BRDF} \downarrow$	Art. $\downarrow$
Loss	Baseline	0.615	0.329	7.570	0.130	0.070	0.325	0.0
	Baseline + PatchGAN	0.810	0.510	3.729	0.089	0.069	0.307	12.0
	Baseline + U-Net D.	0.854	0.658	2.950	0.085	0.068	0.299	28.0
	U-Net D. + $\mathcal{L}_{D_{dec}}^{cons}$	0.858	0.653	2.860	0.088	0.068	0.296	19.0
	$\mathcal{L}_{D_{dec}}^{cons}$ + I_{style}	0.831	0.655	2.790	0.091	0.060	0.289	23.0
	I_{style} + I_{freq}	0.856	0.677	2.410	0.086	0.059	0.288	20.1
Model	I_{freq} + Residual	0.855	0.665	2.310	0.089	0.060	0.285	25.5
	Residual + Decoders	0.863	0.699	2.120	0.079	0.054	0.276	18.5
	Decoders + Attention	0.860	0.665	2.080	0.080	0.059	0.275	0.5
	Attention + ViT	0.863	0.665	2.040	0.079	0.057	0.271	0.5
	ViT + Color	0.899	0.692	1.969	0.068	0.054	0.269	0.0
	Color + Rotations	0.870	0.682	2.050	0.078	0.055	0.271	0.2
Augmentation	Rotations + Distortion	0.876	0.699	2.001	0.074	0.053	0.268	0.0
	Distortion + Erasing	0.893	0.727	1.941	0.067	0.052	0.265	0.0

5.6.3 Uncertainty Evaluation

In Figure 5.8, we show the Pearson correlation matrix between the uncertainty and error for each map, our uncertainty metric (Equation 5.4), and the render perceptual metric (Equation 5.5), on our test set. As shown, neither the error nor uncertainties per map explain errors on the render space. Our proposed σ_{BRDF} achieves a remarkable correlation of 82% with the render error, validating that this metric is useful to predict errors at test time with reasonably high precision.

In the same plot at the bottom, we distill the uncertainty and render error per family, in which we observe that our model struggles to accurately and confidently predict the reflectance of *satins*, *jacquards*, and *leathers* more than it does for any other structure in our dataset. These structures have complex optical behavior that makes their digitization more challenging (for example, satins exhibit anisotropy, which we do not support in our material model), and are relatively uncommon in our training dataset. Figures 5.2 and 5.5 show further examples of our uncertainty estimation for a diverse set of materials.

5.6.4 Active Learning

We leverage σ_{BRDF} for active learning [Sol+21] to identify the samples that contribute most to reduce the error of the model. Figure 5.9 (top) illustrates the process. We start by training a model with 10% of the available training data and measure the uncertainties in the remainder of the dataset. We then select the samples with the highest uncertainties and re-train the model with 20%. We repeat the process with 40, 60, 80 and 100% of the training dataset. For each subset, we compare the performance of this model with four baselines: random-sampling, sampling by the highest uncertainty in normals, roughness, and specular. The results are shown in Figure 5.9 (bottom). Active sampling based on σ_{BRDF} provides significant gains in sample efficiency, obtaining better accuracy for every map. For instance, an actively trained model with our metric which uses 20% of the training data obtains comparable results to a model trained on three times more (but randomly sampled) data. While $\sigma_{\downarrow}$ is typically more informative than σ_{rough} and σ_{spec} , using per-map uncertainties does not provide better results than a random strategy. Finally, Figure 5.2 shows the variation of the probabilistic samples with respect to the ground truth radiance for two materials with high and low uncertainty.

Figure 5.8: Top-left, correlation between render and pixel-wise losses, and uncertainties. Top-right, plot showing the correlation between our uncertainty metric and the error in render space; the renders illustrate the worst and best cases. Bottom, uncertainty and errors for different material families of the test set.

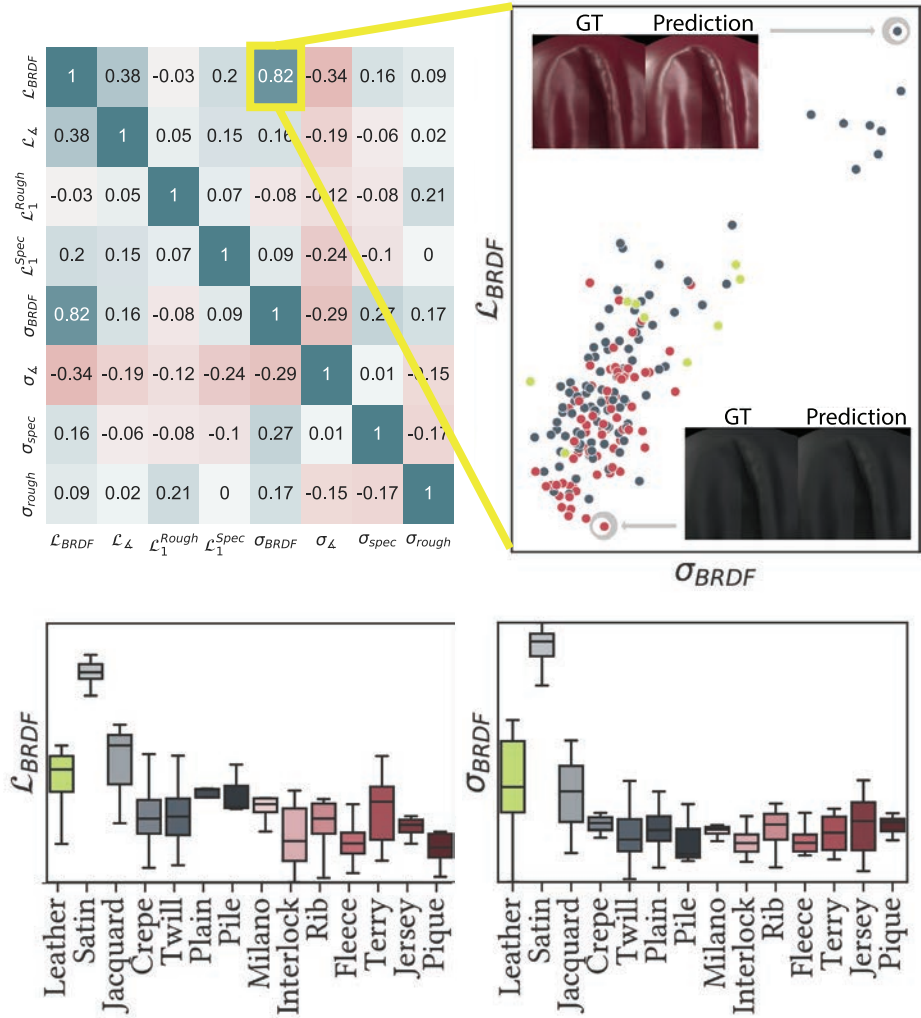


Figure 5.9: On top, illustration of our active learning algorithm. On the bottom, results of our active learning experiments. Leveraging σ_{BRDF} for actively selecting the top-k samples with the highest uncertainty, we achieve better results than a random sampling strategy for every metric we measure.

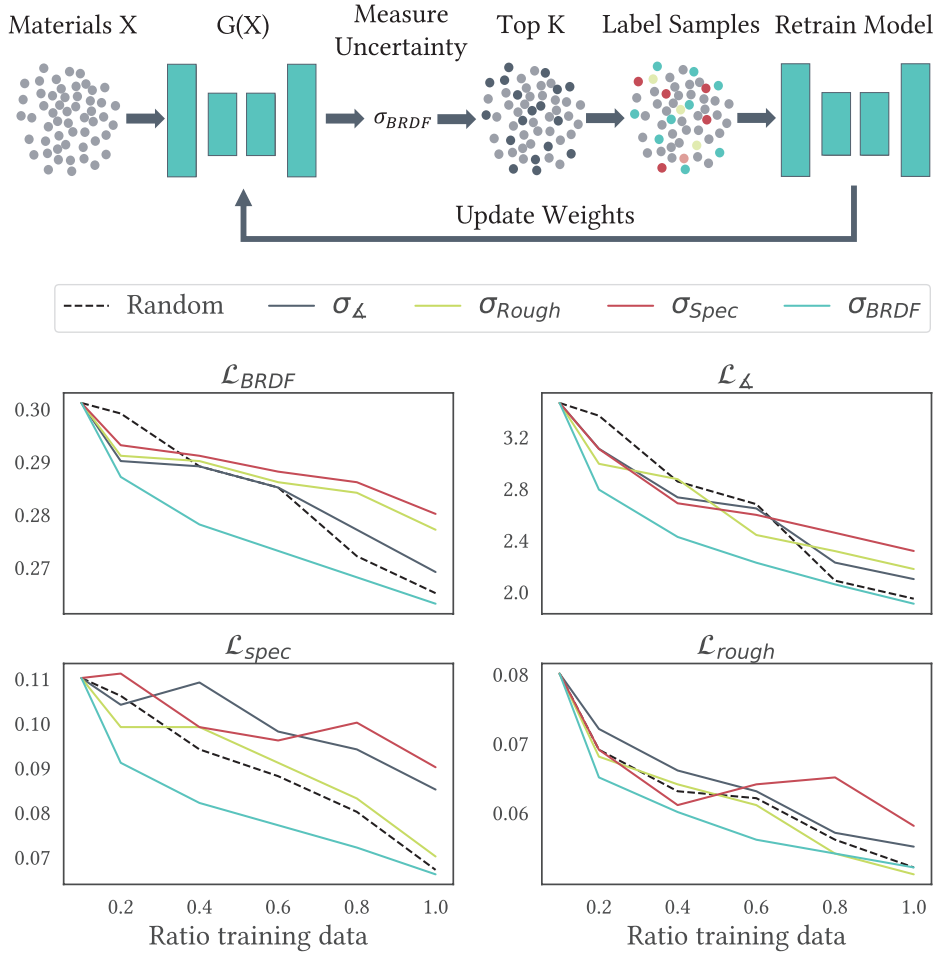


Table 5.2: Model sizes for different methods, and evaluation time in seconds (mean of 100 evaluations on an RTX 2080 GPU), for different output sizes. The methods with * use test-time optimization. DiffMat [Shi+20] does not use a pre-trained model.

Method	Size (MB)	Output Dims	Eval Time (s)
Deep Inverse Rendering* [Gao+19]	167	256×256	603.5
Generative Modeling* [Hen+21]	1095.3	512×384	218.8
Diff. Material Graphs* [Shi+20]	-	512×512	1209.8
Adversarial Estimation [ZK21]	11552.2	256×256	0.078
UMat (Ours)	22.6	256×256 512×512	0.036 0.131

5.6.5 Comparisons with Previous Work

In Table 5.3, we compare our method with previous work on single image material capture. First, we made sure that the training data for these methods included textile materials similar to the ones we choose for testing. Emulating their capture conditions, we took images with a smartphone, with flash and ambient lighting. Note that these capture conditions are not ideal for our method, affecting the final renders if the albedo has shading gradients. However, our goal in this experiment is to evaluate the overall preservation of the material structure in the inferred maps, particularly visible in the normals.

For Shi *et al.* [Shi+20], we initialize the graph using a fabric material, provided in their open-source implementation and include the metallic map. Our model provides sharper and more accurate estimations without requiring optimization during test. Methods trained on style losses [Hen+21; Shi+20] degrade the semantic structure, while Zhou *et al.* [ZK21] generate similar albedos to ours (note that ours are captured), but provide over-smooth estimations. We provide a comparison of timings and model sizes in Table 5.2. With our efficient model design, we can provide real-time estimations without needing any optimization, which also enables our sampling-based uncertainty quantification.

Table 5.3: Comparisons of our method with previous work on images captured under different lighting conditions. Top: a smartphone flash-lit image. Middle: a smartphone image with ambient light. Bottom: a flatbed scanner capture. Our method produces the best results preserving the microstructure even when capture conditions degrade due to the sensor resolution. Note that we do not estimate albedos and that absolute intensities for specular and roughness maps are not directly comparable due to differences in the material model.

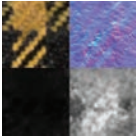
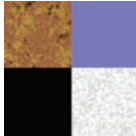
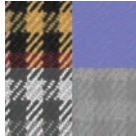
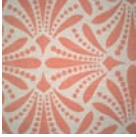
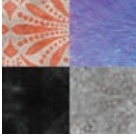
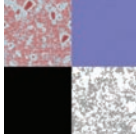
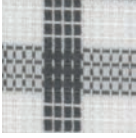
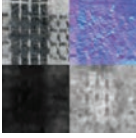
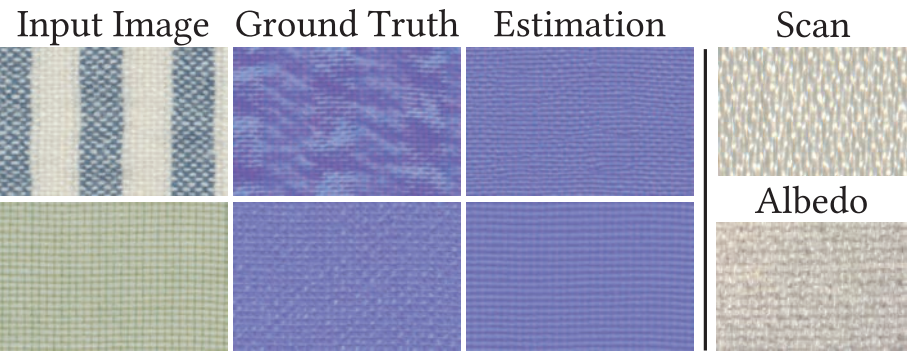
Input	Deep Inverse R. [Gao+19]	Generative Model [Hen+21]	Diff. Material Graphs [Shi+20]	Adversarial Est. [ZK21]	UMat (Ours)
					
					
					

Figure 5.10: Limitations cases of our method. On the left, we show a *seersucker*, with wrinkles that are hidden by the diffuse illumination of the device, and a translucent *organza* with holes between the yarns that appear very bright due to the white background of the scanner, and are therefore mistakenly treated as yarn centers. On the right, we show that for highly directional materials, such as *satins*, the diffuse-like illumination in our capture device sometimes introduces specular highlights.



5.6.6 Limitations

We show some limitations in Figure 5.10. The illumination in the scanner hides the wrinkles in the *seersucker* fabric, and our model predicts a flat surface. The *organza* fabric at the bottom is very transparent with visible holes between the yarns. Since the background is white, the model has mistakenly treated the light regions as yarn centers. For the *satın* at the right, the scan image exhibits specular highlights due to the directionality of the yarns. While this image may be problematic to use as an albedo, it does not affect our metrics as we use constant albedos to compute them.

5.7 Conclusions

We have presented a GAN-based method to digitize materials which leverages microgeometry appearance and a flatbed scanner as a capture device. Our method has shown better performance than state-of-the-art solutions that require a single image as input, when it comes to textile materials. To account for potential ambiguities derived from the capture setting, we have presented a method to model the uncertainty in the estimation at test time.

Managing uncertainty in machine learning projects is important to guarantee robust and functional solutions. However, this typically comes at the cost of complex or slow models. In this work, we have presented the first method to quantify uncertainty in single image material digitization, while introducing minimal impact in the training and evaluation processes. While it is currently not possible to discern the source of the uncertainty, whether it is epistemic (uncertainty which can be reduced by increasing the dataset size) or aleatoric (which is derived by a noisy data generation process), our metric has proven useful to identify ambiguous inputs, underrepresented classes, or out-of-distribution data.

We could extend our work in several ways. The most obvious extension is to estimate real albedos, so that we can deal with other types of scanning devices. Further, expanding our material model to give support for more reflectance properties, such as transmittance or anisotropy, could be useful to improve the realism in the render of textiles.

5.A Additional Implementation Details

5.A.1 Model Design

Our model is trained using a GAN framework. In this section, we detail our design choices for the generator and discriminator architecture.

Generator For the generator, we use a U-Net [RFB15] model, with a few modifications designed to maximize its efficiency, robustness, and generalization capabilities. We specify the full model architecture and layer sizes in Figure 5.11. We use residual connections [He+16; Dia+20; DLG21] in every convolutional block of the model, for better training convergence and preserving details present in the input images. We use 1×1 convolutions on the skip connections. Further, to maximally preserve the appearance and characteristics of every target map, we use a single decoder for each. This has been proposed in different applications, including intrinsic images, material capture, and texture synthesis [Gar+22; RG22; DLG21; ZK21]. To improve the results and enable our uncertainty metric, we append a pixel-wise MLP to each decoder in the model, with Dropout [Sri+14] regularization. Using MLPs after the decoders has been previously explored for material capture [Des+19]. We use Group Normalization [WH18] (with 16 *groups* per layer) and SiLU [EUD18] non-linearities throughout the model. Each convolutional block in the encoder is enhanced with a lightweight Linear Attention module [Wan+20b], with 4 *attention heads*, each with a dimension of 32 hidden units. On the bottleneck, we use a lightweight *MobileViT* Transformer block [MR21], with 128 hidden dimensions for the self-attention and MLPs, 4 layers, and a kernel size of 3. We use a Dropout [GG16] rate of 0.2. We use transposed convolutions for upsampling. Every other implementation detail in the model (strides, bias, poolings) follows [RFB15].

Discriminator For the discriminator [SSK20], we also use a U-Net [RFB15] model, with a few modifications to improve its performance as a discriminator. We specify the full model architecture and layer sizes in Figure 5.12. As in the generator, we use residual connections [He+16; Dia+20; DLG21] in every convolutional block of the model, for better training convergence and preserving details present in the input images. We use Spectral Normalization [Miy+18] and SiLU [EUD18] non-linearities throughout the model. On the bottleneck, we use a lightweight *CBAM* attention block [Woo+18]. The single-scalar estimation of the discriminator D_{enc} is provided by an MLP with a similar architecture to the *CBAM* module. We initialize the entire model using orthogonal initialization [HXP20]. We use transposed convolutions for upsampling. Every other implementation detail in the model (strides, bias, pooling) follows [RFB15].

5.A.2 Model Training

Optimization We train the models using PyTorch [Pas+19] and TorchVision [MR10]. We leverage Kornia for data augmentation [Rib+20]. To accelerate the training process, we leverage mixed precision training and automatic gradient scaling [Mic+18], and train the whole model natively on GPU. Optimization is done using Adam [KB15]. Following [SSK20], we use different learning rates for the generator $lr = 0.001$ and the discriminator $lr = 0.005$ and a batch size

of 10, betas= (0.9, 0.99) and a weight decay of 0.000003. We train the models for 100 epochs, which takes 10 hours on an NVIDIA RTX 3060 GPU.

Loss Function We use the following weights for the loss function: $\lambda_s=3$, $\lambda_{spec}=1$, $\lambda_{rough}=1$, $\lambda_{adv}=0.2$, $\lambda_{style}=0.25$, $\lambda_{freq}=0.2$, $\lambda_{cons}=0.3$. For the style loss function, we use the *AlexNet* variant of LPIPS [Zha+18], as it provides a lightweight style loss which has shown success on texture transfer [RG21].

Data Augmentation Training is done using random crops of 128×128 pixels. We perform random rescales uniformly on the 300, 1000 PPI range, with bilinear interpolation. We randomly rotate the SVBRDF on the $[-10, 10]$ angle range. We use the algorithm in [RG21] to correctly rotate the normal map. On the HSV color space, we randomly change the properties of the model inputs. We randomly change the hue of the images, selected at random on the whole hue ranges. We also change the saturation, value, and contrast of the input images with factors of $[0.9, 1.1]$, as specified on the Torchvision [MR10] ColorJitter implementation. With a probability of 0.3, we apply random Gaussian noise to the input images $\mu = 0$, $\sigma = 0.10$. With a probability of 0.35, we apply random erasing [Zho+20] to the input images. Half of the time, we apply a random Gaussian blur to the input images $\sigma \sim U(0.1, 5)$. Finally, with a probability of 0.3, we apply cut-mix data augmentation to the discriminator, as in [SSK20].

Model Evaluation We use half precision for model evaluation, which makes the process faster and allows us to generate SVBRDF of larger sizes. For images of very high resolutions (eg $\geq 4000 \times 4000$), we use the *stitching* algorithm in [DDB20], with patch sizes of 2048 and stride of 1024 pixels.

5.A.3 Artifact Detection

The thresholds for each material map and the kernel size for the uniformity metric have been optimized given a set of 102 manually labeled textures:

$$t_1(M_s) = 0.01, t_2(M_s) = 1.41,$$

$t_3(M_s) = 1.33; t_1(M_r) = 0.01, t_2(M_r) = 0.99, t_3(M_r) = 3.12$. The size of the box filter is $s_{\text{Box}} = 0.1275 \cdot \text{DPI}_x$. The thresholds for each material map and the kernel size for the uniformity metric have been optimized given a set of 102 manually labeled textures: $t_1(M_s) = 0.01, t_2(M_s) = 1.41, t_3(M_s) = 1.33; t_1(M_r) = 0.01, t_2(M_r) = 0.99, t_3(M_r) = 3.12$. The size of the box filter is $s_{\text{Box}} = 127.5$.

5.A.4 Capture Details

We construct the training and testing dataset capturing 10×10 cm samples at 1000 PPI using an *EPSON V850 Pro* on its default settings.

For the comparisons with previous work, we place the fabric samples on a black surface. We use a *Huawei Nova 5T* smartphone, and capture the materials at two distances: a *close-up*, capturing 4×4 cms at a distance to the sample of 8 cms, and a *full-size* image, capturing 8×8 cms, at a distance to the sample of 13 cms. We use ISO=50, an aperture of $\frac{f}{1.8}$, and a focal length of 26mm for every image. For the two distances, we capture the material using ambient illumination, with exposure times of $\frac{1}{10}s$ for the *full size* and of $\frac{1}{8}s$ for the *close-up*. We also capture the images using the smartphone flash lighting, with exposures of $\frac{1}{40}s$ and $\frac{1}{50}s$ for the *full size* and *close-up*, respectively.

5.A.5 Comparisons with Previous Work

We perform every comparison with previous work on an RTX NVIDIA 2080, using the default configuration for every method. However, for [Shi+20], we initialize the material graph with a fabric material (fabric suit vintage) provided in their repository, to better match our test data. For [Hen+21], we use the *fine-tuning* configuration.

Figure 5.11: A full diagram of our generator, including layer sizes and output dimensions for each layer. For Self-Attention, we leverage Linear Attention [Wan+20b], we use a MobileViT transformer on the bottleneck [MR21], Group Normalization [WH18] and SiLU [EUD18] non-linearities, one decoder per output map and residual connections in every convolutional block. In red, we show the input/output dimensions (spatial, channels) of each layer; in orange, we show attention modules; in blue, convolutional blocks and layers; in green, upsampling and concatenating operations; in yellow, normalization layers; and in purple, regularizations, and non-linearities.

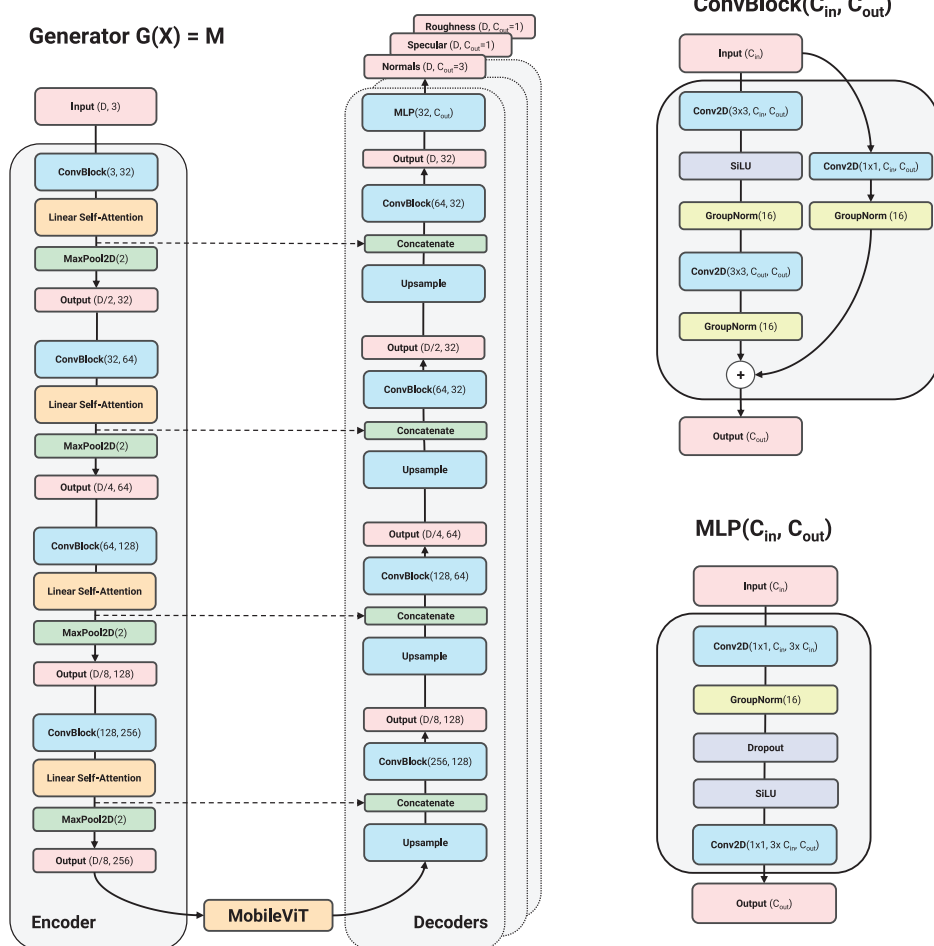
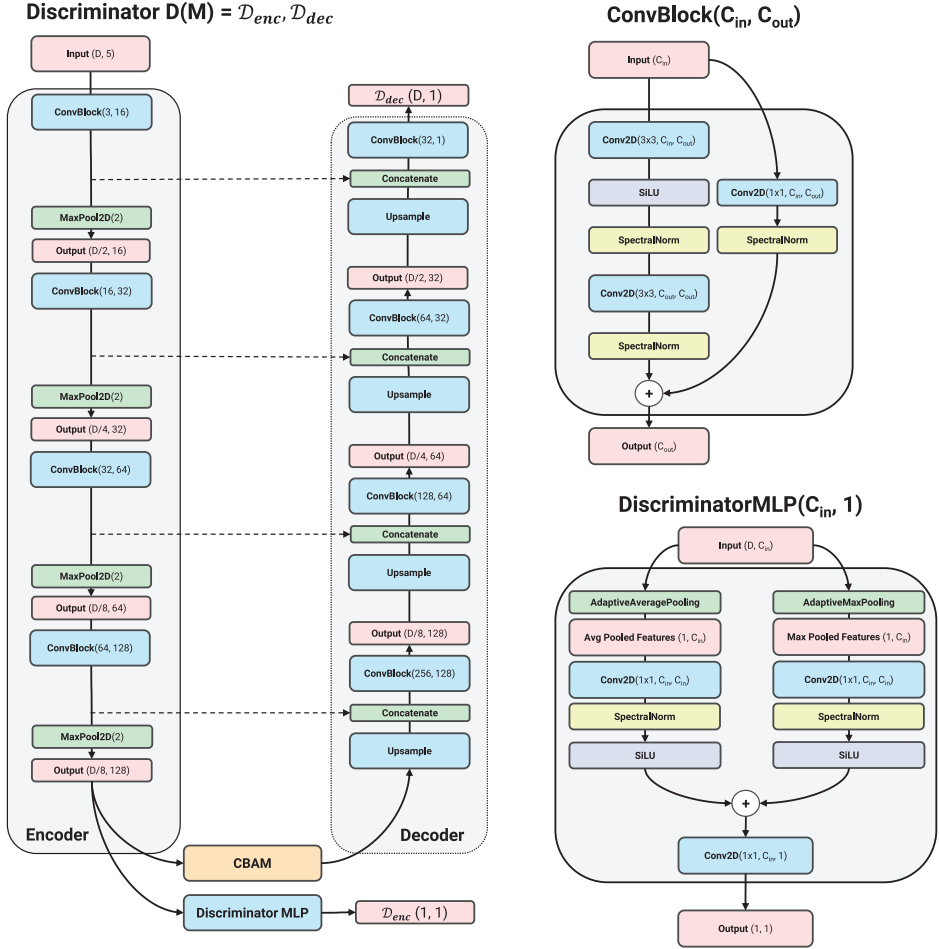


Figure 5.12: A full diagram of our U-Net residual discriminator, including layer sizes and output dimensions for each layer. We use a CBAM [Woo+18] module on the bottleneck and Spectral Normalization [Miy+18] throughout the network and residual connections in every convolutional block. In red, we show the input/output dimensions of each layer; in orange, we show attention modules; in blue, convolutional blocks and layers; in green, upsampling and concatenating operations; in yellow, normalization layers; and in purple, non-linearities.



5.B Dataset Analysis

Figure 5.13: Visualization of our dataset. We show the percentages of materials in our training dataset, including more detailed subcategories. On their right, we show the average specular and roughness for every category. As shown, there are some structures with distinct characteristics: Satins are highly specular due to the particularities of their yarns, and Piles (eg corduroy) or Plain Weave (eg linen fabrics) are much less glossy. We exploit this relationship between microgeometry and specularity for our estimations.

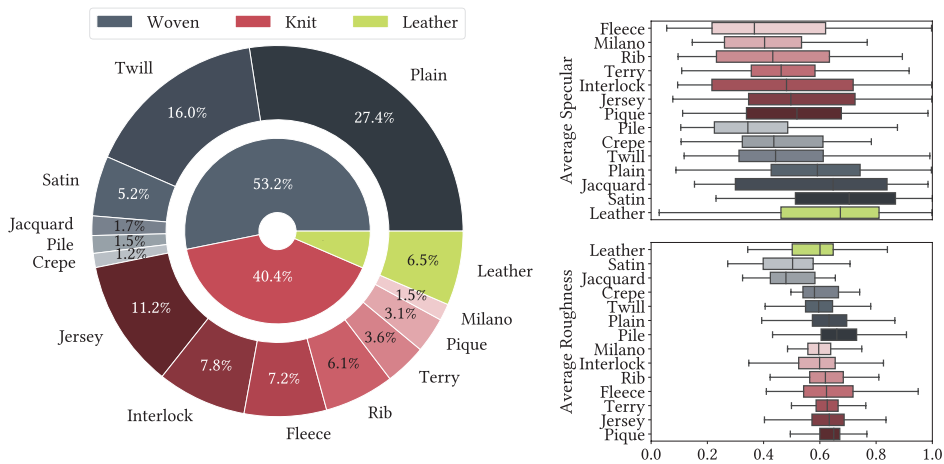
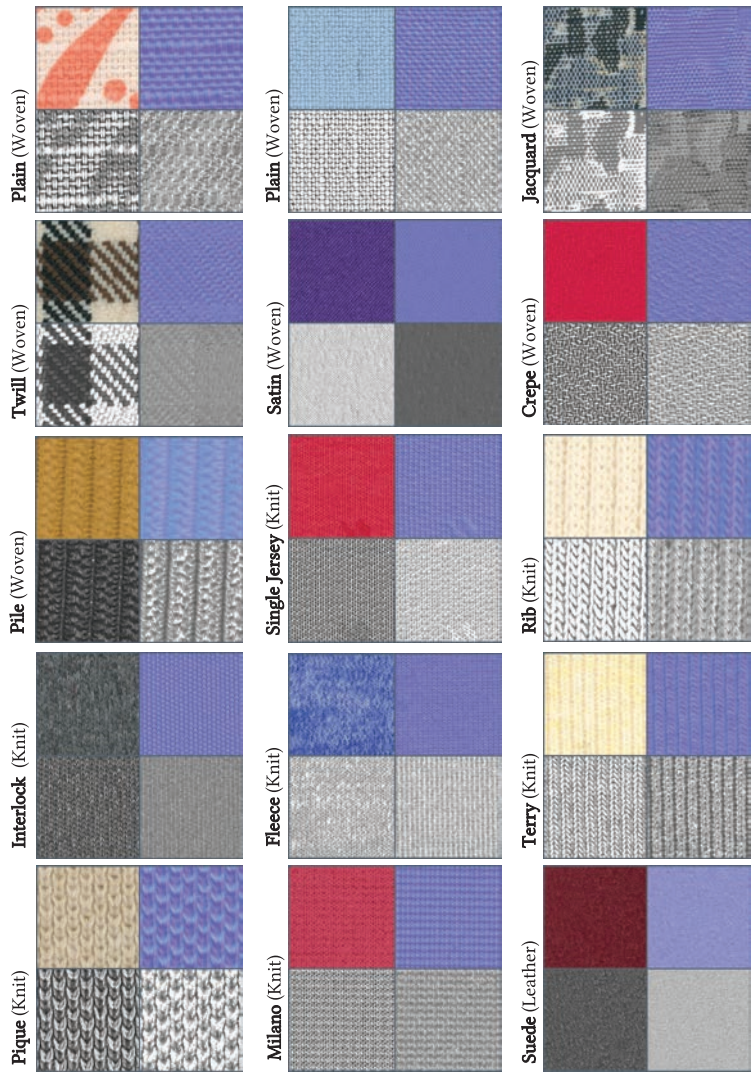


Figure 5.14: Visualization of some Ground Truth SVBRDF of the different families in our test set. Textiles have very complex and varied microstructures which play an important role on their appearance at different scales.



5.C Additional Details And Results

Figure 5.15: Additional results of our uncertainty quantification method. On the top, we show a beige *rib* fabric with gray metallic yarns. While there is a low uncertainty for the beige yarns, the metallic yarns are harder to digitize for our model (we do not support metalness in our material model) and it shows a higher uncertainty on those yarns. In the middle, we show a leather material with a very strong structural pattern. Our model shows very low confidence for this material. In the bottom, we show a tartan fabric. Interestingly, our model shows a higher uncertainty on the blue yarns, which are less common than the other yarns in the material.

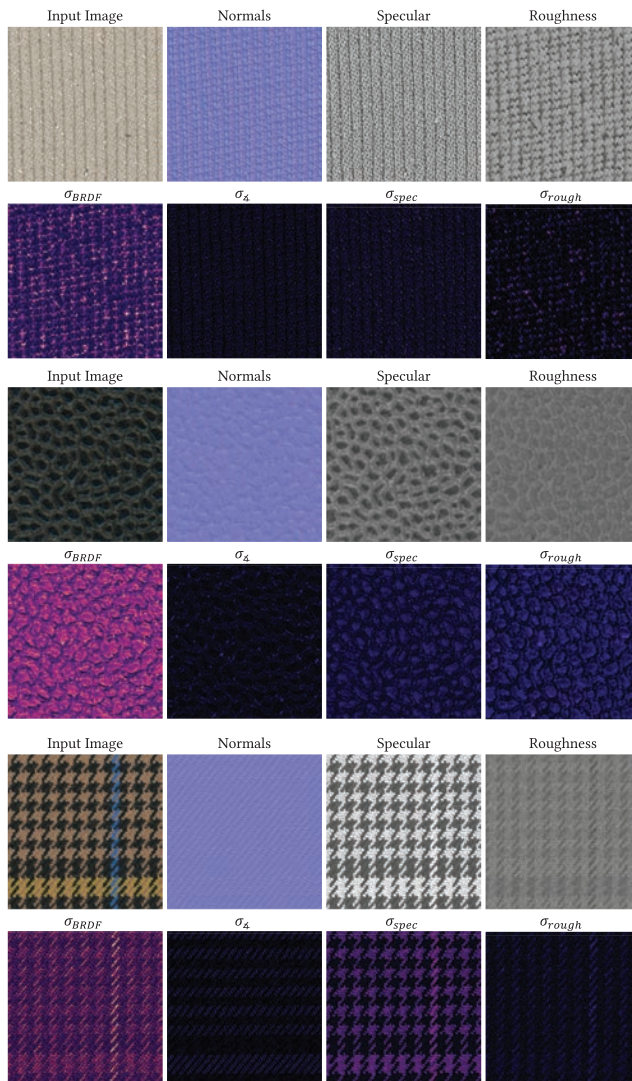


Figure 5.16: Additional results of our uncertainty metric. On the top row, we show a plot between our proposed render uncertainty σ_{BRDF} and the render error, which are highly correlated. We also show average error and uncertainty per family, as well as renders with the lowest and highest errors, compared to the ground truth. Below, we show the same data for normals, specular and roughness errors and uncertainty. There are no correlations between uncertainty and errors for these maps.

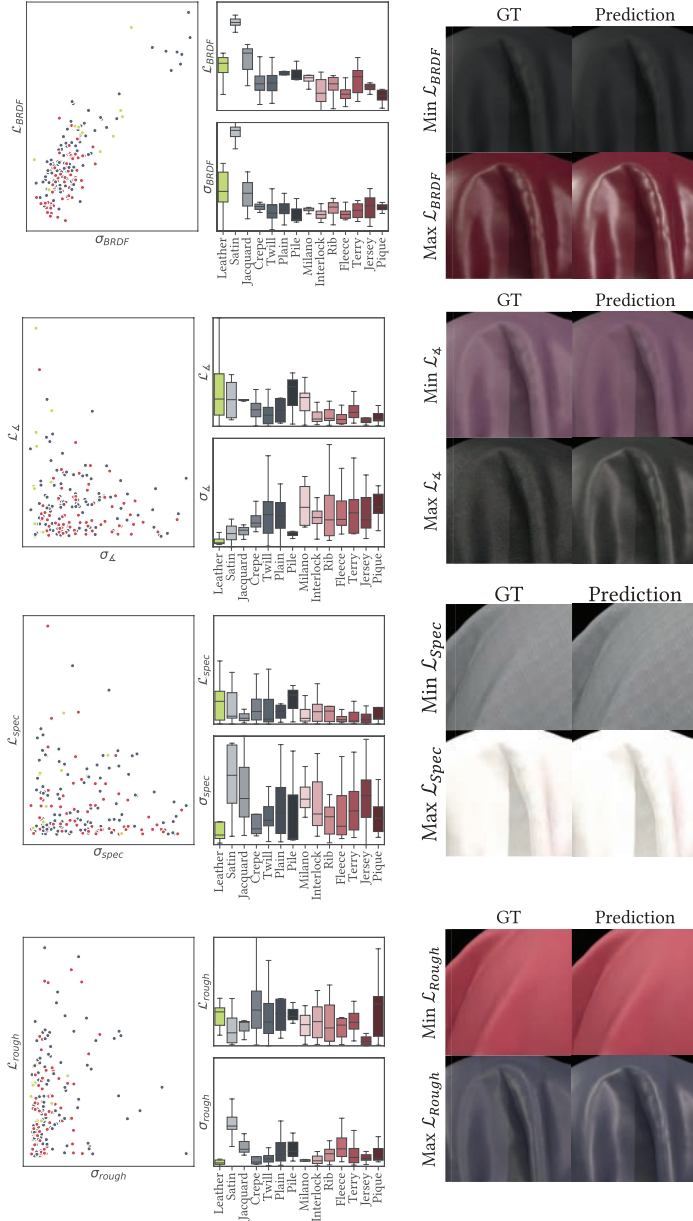


Figure 5.17: Additional results of our active learning experiment. From top to bottom, we show the renders of ground truth SVBRDFs, the model trained on the 100% of the training data, a model trained on 40% of the training data available, selected following an active learning approach using our uncertainty σ_{BRDF} as guidance, and a model trained on 40% data, selected randomly. From left to right, we show a very diffuse *Chiffon* fabric, a highly specular *Shantung* fabric and a *Goat Leather* material with very varied microgeometry. In every case, our model trained on 40% of the data following an active learning approach obtains results which are very similar to a model trained on 100% of the data. The model trained on randomly selected 40% data produces highly inaccurate specularities and microgeometry estimations.

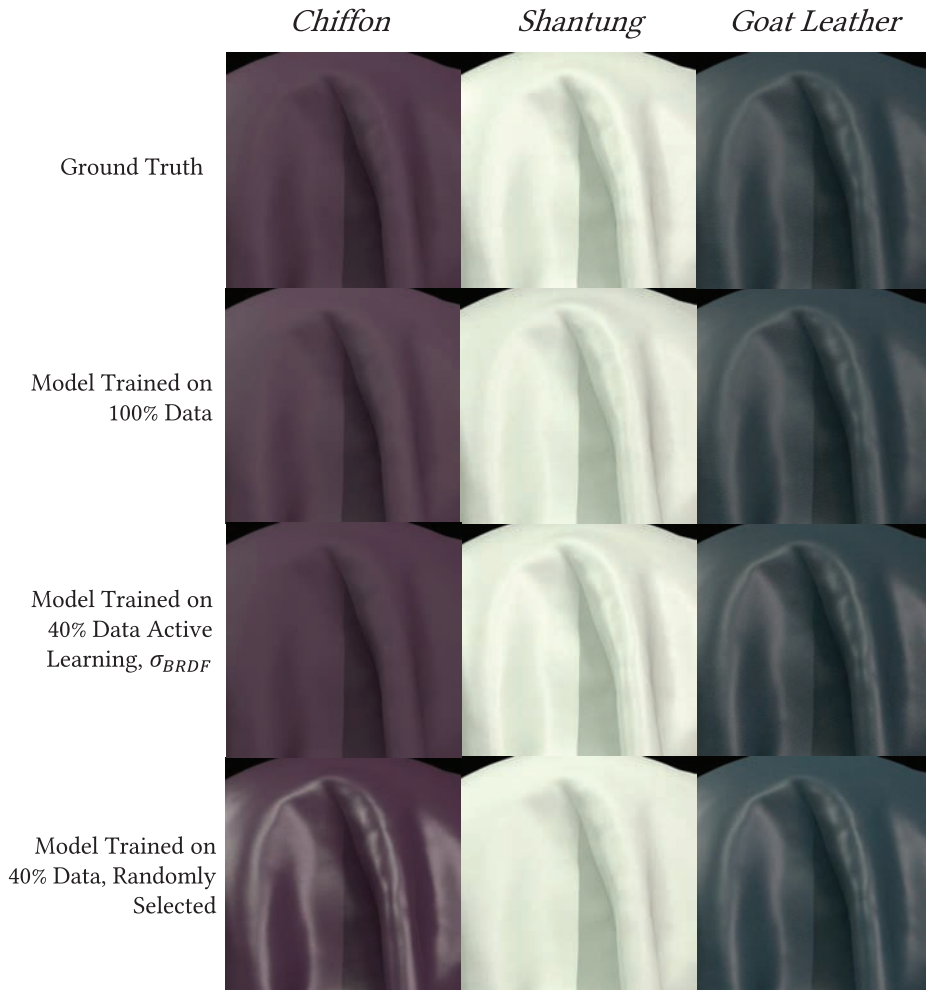


Figure 5.18: Results of our method on datasets of previous work. On the top, we show the results of [Hen+21] on their own test set. Our method provides sharper normals which better preserve the structure of the input images. In the middle, we show the results of our method on synthetic rendered data from a material graph from [Shi+20]. Our method provides sharp normals for this synthetic image, which lies outside the distribution of our dataset, composed exclusively of real images.

On the bottom, we show an albedo and normals computed using Photometric Stereo [Ike81] for a captured BTF [WGK14], for which our model also provides highly detailed results.

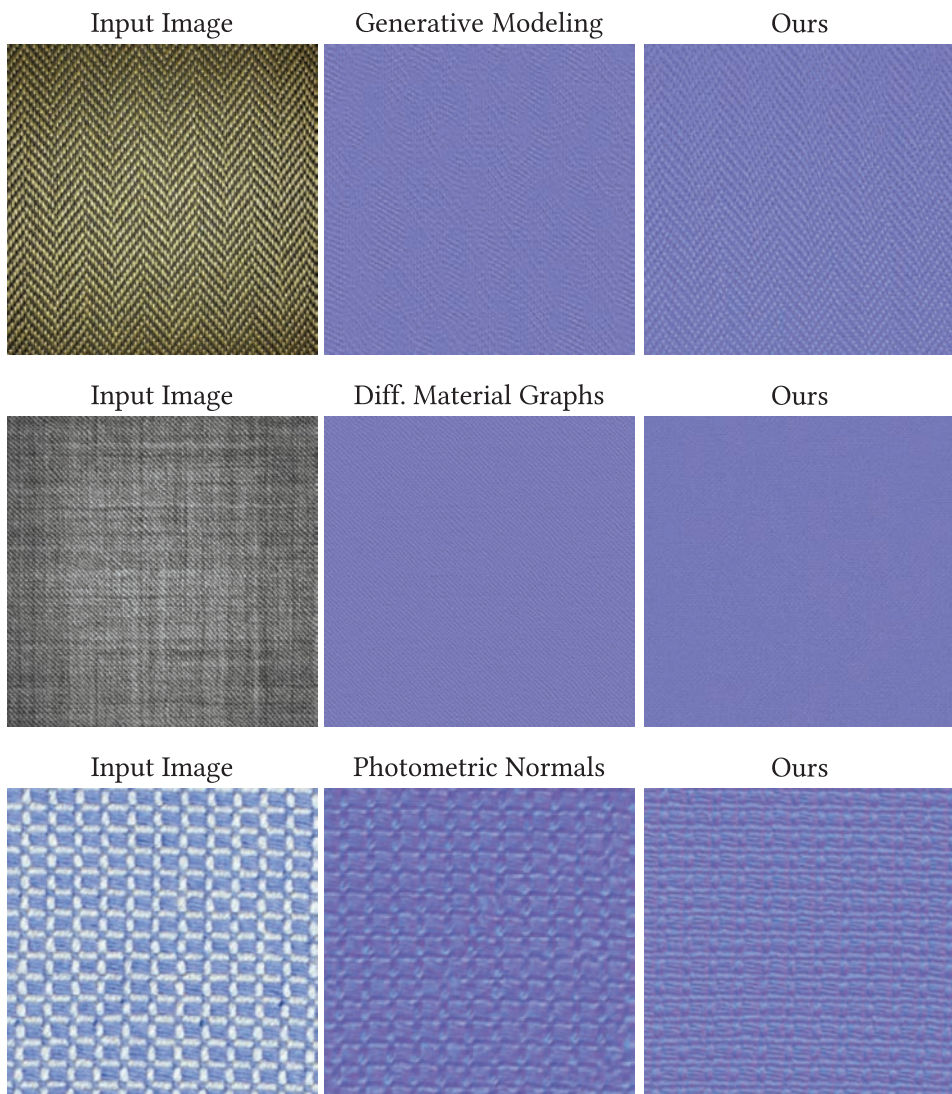


Table 5.4: Comparisons of our results with previous work on images captured under different conditions for a *curdoroy fabric*: On the first rows, images were captured with a smartphone using the flash image. On the middle rows, using the same smartphone with ambient lighting on different scales. On the final row, a scanner image. Note that for [Shi+20] we use a fabric material for initialization and use their metallic map instead of specular, that we do not estimate albedos and that the material models are not necessarily comparable.

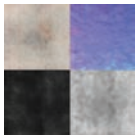
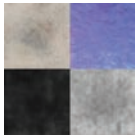
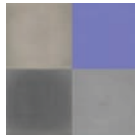
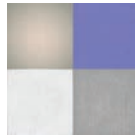
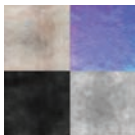
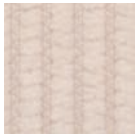
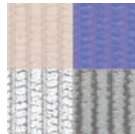
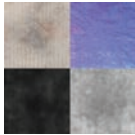
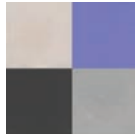
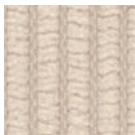
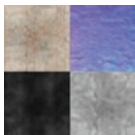
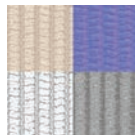
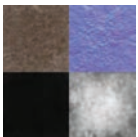
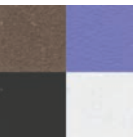
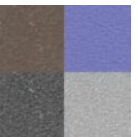
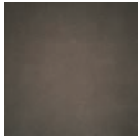
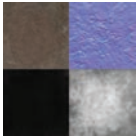
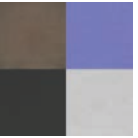
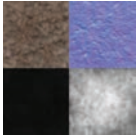
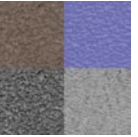
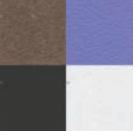
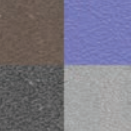
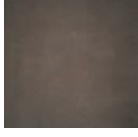
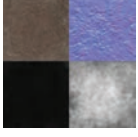
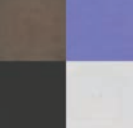
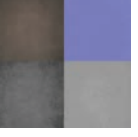
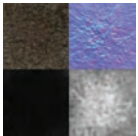
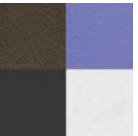
Input	Deep Inverse R. [Gao+19]	Generative Model [Hen+21]	Diff. Material Graphs [Shi+20]	Adversarial Est. [ZK21]	Our Method
					
					
					
					
					
					

Table 5.5: Comparisons of our results with previous work on images captured under different conditions for a *suede leather*: On the first rows, images were captured with a smartphone using the flash image. On the middle rows, using the same smartphone with ambient lighting on different scales. On the final row, a scanner image. Note that for [Shi+20] we use a fabric material for initialization and use their metallic map instead of specular, that we do not estimate albedos and that the material models are not necessarily comparable.

Input	Deep Inverse R. [Gao+19]	Generative Model [Hen+21]	Diff. Material Graphs [Shi+20]	Adversarial Est. [ZK21]	Our Method
					
					
					
					
					
					

CHAPTER 6. CASUAL CAPTURE OF FABRIC MECHANICS

Figure 6.1: From two depth images of a fabric sample casually placed in two specific configurations (input), we infer its corresponding set of mechanical parameters. These can be used in a simulator, to visualize the drape of any garment (output). Our method introduces a perceptual drape similarity metric, which enables sorting materials based on their drape.



In this chapter, we propose a method to estimate the mechanical parameters of fabrics using a casual capture setup with a depth camera. Our approach enables to create mechanically-correct digital representations of real-world textile materials, which is a fundamental step for many interactive design and engineering applications. As opposed to existing capture methods, which typically require expensive setups, video sequences, or manual intervention, our solution can capture at scale, is agnostic to the optical appearance of the textile, and facilitates fabric arrangement by non-expert operators. To this end, we propose a sim-to-real strategy to train a learning-based framework that can take as input one or multiple images and outputs a full set of mechanical parameters. Thanks to carefully designed data augmentation and transfer learning protocols, our solution generalizes to real images despite being trained only on synthetic data, hence closing the sim-to-real loop. Key in our work is to demonstrate that evaluating the regression accuracy based on the similarity at parameter space leads to inaccurate distances that do not match the human perception. To overcome this, we propose a novel metric for fabric drape similarity that operates on images instead of the parameter space, allowing us to evaluate our estimation within the context of a similarity rank. We show that our metric correlates with human judgments about the perception of drape similarity, and that our model predictions produce perceptually accurate results compared to the ground truth parameters. The contributions in this chapter led to the following publication¹:

¹ Additional publication details and results are included on the project website.



"How Will It Drape Like? Capturing Fabric Mechanics From Depth Images"

Carlos Rodriguez-Pardo, Melania Prieto-Martin, Dan Casas, Elena Garces

Computer Graphics Forum (Proc. Eurographics 2023) (2023)

6.1 Introduction

Creating accurate digital representations of real-world materials, or *Digital Twins*, is crucial for enabling realistic 3D visualizations suitable for interactive design and predictive engineering. Some industries, like fashion or textile manufacturing, further require these methods to work at a scale to cope with the fast pace of the current production workflows. However, digitizing cloth is challenging due to the high variability and type of fabric samples, where the fabric composition, the microstructure, or the finishing play crucial roles in the perceived appearance.

While casual systems which obtain optical appearance have long been a focus of research, comparably less attention has been paid to estimating *mechanical* properties. Indeed, capturing and simulating the mechanical behavior of cloth is challenging due to the complex interplay between the internal and external forces occurring in this type of physical system, which is highly sensitive to the environmental conditions. Nevertheless, with the current need to create virtual copies instantly, a casual setup able to produce automatic and accurate estimates—beyond having a set of *presets* from which to manually choose the closest one—of mechanical parameters could prove very valuable.

Previous methods are impractical for scalable and customizable workflows. Accurate fabric parameter acquisition systems require specialized and expensive devices [Kaw80; Min95; Cla+90], which are often slow and need skilled operators. Existing casual capture setups use input video sequences [Bha+03; Bou+13; YLL17] or, even if they take a single image, might require manual user input [JC20]. In this chapter, we present a casual capture system that only requires taking two depth images of the textile posed in a static drape. Our capture setup does not require complex calibration, can be easily manipulated by non-expert operators, and is agnostic to the optical properties of the fabric

thanks to leveraging depth images instead of RGB data. Figure 6.2 illustrates our capture setup. It involves capturing the fabric with a depth camera in two relaxed positions: the *hanging* scene, which conveys the drape when no force other than gravity is applied to it, and the *stretch* scene, which provides cues on the stretching properties of the fabric.

We propose a learning-based method to instantly return the mechanical parameters given static depth images and the fabric density as input. Our method relies on a sim-to-real strategy [Zhu+21], leveraging transfer learning and building on a dataset of physical fabrics digitalized with precision equipment. Our model is trained solely with synthetic data, and thanks to carefully designed policies of data augmentation and neural features, it can generalize to real-world scenarios. Under the hood, our approach leverages a custom architecture that enables a flexible design, that can take one or multiple images as input, enhancing its performance when more data is available. We perform an extensive evaluation by means of ablation studies and by measuring aggregated neural network saliency maps, which show that some scenes are more informative than others for predicting the mechanical properties of the fabric. Furthermore, we demonstrate the performance of our model on real-world captured samples, showcasing our system’s generalization capabilities.

Figure 6.2: Capture setup, RGB (top), and depth (bottom) images for *hanging* (left) and *stretch* (right) scenes in rest position. Each scene conveys a different mechanical appearance of the fabric: *hanging* exhibits the overall drape; *stretch* exhibits an extra diagonal tension, which is key to understand the stretching properties.



Key to our work is to demonstrate that evaluating the prediction accuracy of mechanical properties using typical error metrics, such as the Mean Absolute Error (MAE) on the parameter space, leads to inaccurate distances that do not

match the human perception. We identify that such mismatches occur due to two factors: first, the parameter space is not bijective—i.e., different sets of parameters might convey the same drape—; and second, a numerical error in a parameter does not necessarily correlate with what we perceive as an error. To address this shortcoming, common in all existing works, and inspired by previous work on similarity metrics for material appearance [Lag+19], natural images [Zha+18] or illustration [Gar+14], we propose an image-based metric that measures differences on the mechanical behavior of textiles taking into account the overall drape. We validate that our metric agrees with human perception, and it can be used to sort materials by drape similarity with respect to a reference fabric.

Using our similarity metric, we finally validate that the estimations of our method correlate with human judgments about drape similarity, and that our model predictions produce perceptually accurate results compared to the ground truth parameters. All in all, our approach makes an important step towards solving sim-to-real problems for mechanical estimation since it shows that simulated cloth using inferred parameters maximizes the similarity with respect to real-world target fabrics.

6.2 Related Work

6.2.1 Parameter Estimation Methods

Estimating the mechanical properties of real fabric samples is a highly challenging problem for several reasons: the number of uncontrollable extrinsic factors (e.g., wind forces, initial state, collisions, etc.) which affect the predictiveness of the physical simulation; the lack of a standard deformation model and parameter spaces; and the use of computation-intensive simulation methods. Accordingly, a wide range of strategies exist, aimed at overcoming these challenges.

Measurement Devices. It is common to combine optimization techniques with the output of testing devices to find the optimal set of parameters which best explain the observations [Mag+07; SB08; VMF09; WOR11; Mig+12; CTT17]. Existing technologies of this type are diverse and, as discussed by Kuijpers *et al.* [KLG20], lack of a clear standard. The Kawabata Evaluation System (KES) [Kaw80] is perhaps one of the most well known, measuring 16 coefficients including bending, shearing, and tensile among others. Despite its precision, this method was not widely adopted by the industry due to its lengthy processes and the need for expensive equipment. Consequently, several other methods tried to simplify and unify the methodology with partial success

according to some studies [LM08; Pow13]: the Fabric Assurance by Simple Testing (FAST) [Min95], the Fabric Touch Tester (FTT), the CLO Fabric Kit 2.0, the Fabric Analyser by Browzwear (FAB), the Optitex Mark 10, and the cantilever principle [Cla+90].

Reconstruction-Optimization Methods. Another set of techniques jointly tackles the reconstruction and parameter optimization problems. By taking as input data from arbitrary real simulations (e.g., the cloth deforming on an avatar [Yan+18]), they iteratively reconstruct and simulate the scene which better explains the observation. Bhat *et al.* [Bha+03] takes as input a video sequence of the cloth and uses simulated annealing to optimize the parameters by measuring its folds. Yang *et al.* [YL15] use multi-view stereo reconstruction to initialize the 3D shape. Runia *et al.* [Run+20] also introduces simulation steps to explain observed phenomena of cloth in the wind. They rely on similarity metrics computed on deep latent spaces to supervise the optimization of the parameters. These methods require simulation steps embedded into the fitting processes, making them computationally expensive due to the high dimensionality of the parameter spaces. Recent differentiable simulation techniques [LLK19; JBH19; Hu+19; Mur+20; Li+22] have proven to be efficient ways to reduce the fitting burden by enabling the computation of gradients with respect to these parameters within latent spaces of neural networks, taking into account dynamics, self-collisions, and contacts.

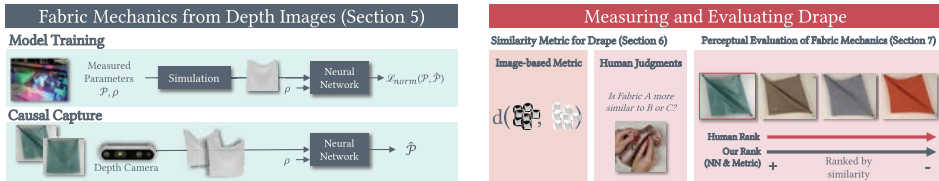
Data-Driven and Regression Methods. The third set of methods avoids reconstructing the original 3D scene by working on an estimated feature space and leveraging previously simulated data and machine learning techniques. Our approach falls into this category. Taking videos as input has been explored by Bouman *et al.* [Bou+13] to recover stiffness and area weight using a descriptor of the image based on PCA and optical flow, and later by Yang *et al.* [YLL17], who leverage neural networks to extract image features used for regression. Davis *et al.* [Dav+15] estimated the same simulation parameters by exploiting imperceptible vibrations in high-speed video recordings. Bi *et al.* [Bi+18] further evaluated that humans also need fabric motion to understand its stiffness. Friction coefficients have been estimated using reflectance values [ZDN16] or dynamic videos of cloth sliding through a surface [Ras+20]. Instead of regressing the parameters, Huber *et al.* [HEW17] find the most similar cloth in a database using motion descriptors. A different approach only using a single image of the *Cusick drape* was followed by Ju *et al.* [JC20], but it requires a 360° scan to reconstruct the target cloth, and a manually fitted Bezier curve to obtain the feature vector. In contrast, we just require a depth map that can be captured easily. Concurrent work [Fen+22] uses multiple-view depth images as input to a trained regressor.

Our approach is inspired by these ideas; however, we do not require optimization—providing instant estimation of the parameters— and leverage neural features to understand and model fabric behavior in a semi-controlled setup. We demonstrate that our approach works with two images as input to predict bending and stretching coefficients without requiring a full video of the piece of fabric.

6.2.2 Pre-Trained Models and Transfer Learning

Deep learning models typically require vast amounts of data for generalizing to unseen examples. When this amount of data is not possible to acquire, *transfer learning* techniques help by re-using model parameters trained on a related task [KSL19; Rag+19; Bom+21]. These techniques include: *Fine-tuning* the weights of a pre-trained classification model [WKW16; RD17; Che+18; Wan+18; RJ19; Guo+19; Kol+20]. Pre-training an image descriptor model on contrastive or self-supervised learning tasks, and uses the activations of its last layer as input to the downstream task (*Linear Probing*) [Che+20b; Rad+21; CXH21; He+22; Kum+22]. For domain adaptation problems, it is common to *adapt* the internal representations of pre-trained CNNs so as to efficiency [RBV17; RBV18; RB19; Pha+20; LLB22]. Inspired by these approaches, we design a model that leverages fine-tuning of a pre-trained image CNN classifier as a feature extractor, capable of processing depth images, and extending it to account for additional input variables, and handling multiple images at the same time during inference.

Figure 6.3: An overview of the main components of our method. We propose a system to estimate fabric mechanics using depth images of *hanging* and *stretch* scenes as input. To validate the error of our estimations perceptually –accounting for the global drape–, we propose an image-based drape similarity metric which we validate with human judgments and can be used to sort fabrics by similarity. We show through several metrics that the estimations provided by our method using our similarity metric agree with those given by humans.



6.2.3 Similarity Metrics

Full-Reference Image Quality Assessment (IQA) aims to provide a single score that measures the amount of distortion between two images. Traditionally, these metrics leveraged low-level image statistics. *PSNR* is commonly used for measuring image degradation, but correlates poorly with human perception [Zha+18]. More sophisticated alternatives have been developed, including *SSIM*, *mSSIM* [Wan+04], and others [Zha+11; Naf+16; ZSL14; ZL12; Rei+18]. Algorithms based on latent spaces of CNNs [GEB16a] have been extended to better approximate human perception, for example, by training on a large pool of human evaluations *LPIPS* [Zha+18], or by other means [Din+20; Pra+18].

Besides, similarity metrics that measure abstract or complex concepts like style have been proposed for 3D furniture [LKS15], illustration [Gar+14], icons [LGG19], product design [LKS15], or material appearance [Lag+19]. Unlike ours, these metrics require to be trained with human ratings, thus incurring a considerable cost to collect such information via user studies. Instead, our metric does not require specific training, leveraging an off-the-shelf image-based metric. Despite this, we show that our metric correlates with human judgments on the perception of fabric drape similarity, and that can be used to evaluate the overall drape.

6.3 Overview

Figure 6.3 presents an overview of our work. First, in Section 6.5, we introduce our novel solution to infer fabric mechanics directly from depth images. As input, our approach only requires static depth images in two specific configurations, shown in Figure 6.2, as well as the fabric density which can be easily obtained with conventional equipment. In Section 6.4 we describe the datasets of *synthetic* and *real* samples, with mechanical ground truth parameters, used train and evaluate our regressor.

Figure 6.4: Spearman correlation matrix between parameters of our synthetic dataset.

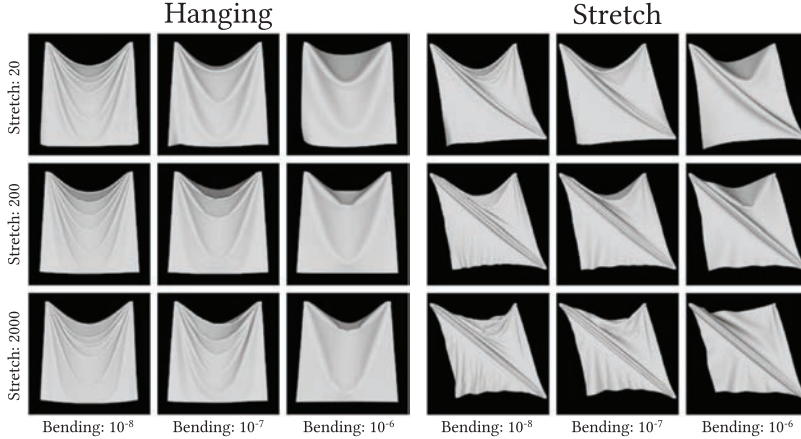


The quantitative evaluation suggests that our method is able to estimate the mechanics within a certain error. However, since a direct interpretation of that error is not human-friendly, we propose a method to evaluate the overall drape in the context of a real scene. In Section 6.6, we introduce our image-based similarity metric for drape, which takes as input renders of the chosen scenes and provides a relative value that is useful to compare the drape of different fabrics. With a user study, explained in Section 6.7.1, we validate that our metric agrees with human preferences on the global perception of fabric drape. Also, in Section 6.7.2, we evaluate our capture method using our drape similarity metric and compare it with human judgments. We effectively validate that our estimations agree with human assessments and provide several qualitative examples in Section 6.7.3.

6.4 Datasets

We develop two different datasets of depth images, which we use at different steps of the pipeline to train and evaluate our models: a *synthetic* dataset, generated using physics-based cloth simulation; and *real* dataset, generated using images of real fabric samples. In both datasets, a sample consists of a depth image of a fabric simulated in a scene for which we have the corresponding mechanical parameters, namely: bending [Gri+03] and stretch [VMF09] in the warp, weft, and bias directions and the fabric density, $\{kStretchWarp, kStretchWeft, kStretchBias, kBendingWarp, kBendingWeft, kBendingBias, \rho\} \in \mathbb{R}^7$. We support two different static configurations for the scenes: *hanging*, which exhibits the overall drape; and *stretch*, which exhibits diagonal tension and is key to understanding the stretching properties. Figure 6.2 depicts examples of each configuration.

Figure 6.5: Sweep of simulation parameters for hanging and stretch scenes. For $kBending$: warp, weft, and bias have the same value, while for $kStretch$, bias changes as 100, 144, 1000.



Simulated Dataset. To train our model, we generate a synthetic dataset by simulating fabrics in a virtual scenario, replicating the *hanging* and *stretch* configurations. We use a standard simulator, similar to ARCSim [NSO12], with a quadratic strain, a linear strain/stress relationship, and standard definitions for bending and stretch [Gri+03; VMF09]. To model highly anisotropic fabrics, we use three parameters for warp, weft, and bias. Note that the model used for stretch already has some nonlinear behavior (quadratic strain), but more parameters (one per direction) are required to control the nonlinearity of the forces. Thickness is not included as it is implicitly accounted for in the other parameters. Contrarily, density is required as the simulator is dynamic. In a static one, it could be dropped after normalizing the other parameters.

After simulation, we render the resulting mesh (discretized at 5mm/edge) with a white Lambertian material and extract the depth buffer. To ensure that our synthetic dataset covers a wide range of materials, we densely sample the parameter space using a distribution of common mechanical parameters of real-world fabrics. Figure 6.5 shows a sweep of parameters showcasing the variability of resulting drapes in the hanging and stretch scenes. To better understand potential relationships between the parameters, we compute the Spearman correlation r , shown in Figure 6.4, where we observe the higher correlation between the three $kStretch$ coefficients and some correlation between the $kBendingBias$ and the density.

Real Dataset. To evaluate our model, we test it with real data from images captured by the Intel RealSense SR300. To this end, we casually hang 50×50 cm fabric samples using several magnets into a metallic panel, which

requires little to no expertise and can be done very fast. Pin location does not need to be centimeter-accurate. Because we use depth images, no special lighting is necessary. We measure fabric area density by weighing a 10×10 cm sample and dividing by its area. See Figure 6.2 for a visualization of our capture setup and the accompanying video for an illustration of the process. We capture ten fabrics of diverse compositions and structures, for which we have ground truth mechanical parameters previously obtained with specific equipment and methods [Spe+22].

6.4.1 Data Augmentation

To make our model robust to potential noise and scenario variations common in uncontrolled capture setups (e.g., slightly different camera viewpoint or fabric configuration), we apply several data augmentation strategies to our *synthetic* dataset.

Simulation-Space Data Augmentation. To enforce robustness to different camera view-points, simulated meshes are rendered within a range of different inclinations with respect to the vertical plane. This range covers ± 5 degrees from the rest position orientation, creating 11 depth maps per material and scene.

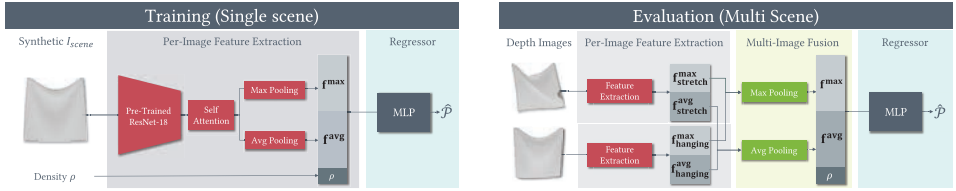
Image-Space Data Augmentation. Real images are largely different from synthetic images, due to distortions, noise, perspective changes, unknown illumination, lens and sensor characteristics, blurs, etc. To enforce the robustness of our models to those distortions, we design an extensive image data augmentation policy consisting of random individual deformations, performed in a particular order. These not only include random noise, blurs, perspective changes and rescales, but also more complex policies such as thin-plate deformations, posterization and erasing. This data augmentation policy bridges the gap between synthetic renders and real depth images, which are typically noisier.

6.5 Fabric Mechanics from Depth Images

In this section, we present our learning-based approach to estimate fabric mechanical parameters $\hat{P}$ from depth images. Given a set of depth images, $I = \{\text{hanging}, \text{stretch}\} \geq 1$, which depicts the *hanging* and *stretch* scenes, we train a model $\mathcal{M}$ which maps I , along with the material *density* ρ , to mechanical parameters: $\mathcal{M}(I, \rho) = \hat{P} \in \mathbb{R}^6$. To this end, at train time we learn to extract relevant features from depth images, which are then fed into a regressor to learn to

predict mechanical features. Importantly, our architecture enables to input sets of images at test time by fusing their respective features. Figure 6.6 illustrates the train and test pipelines.

Figure 6.6: Diagram of our training and evaluation pipelines. For **training**, we use a single image of the material, along with its density. The image is processed by our *Feature Extractor*, followed by an MLP, which computes the parameter estimation $\hat{\rho}$. For **evaluation**, we process each available image with our feature extractor and use a *fusion* operator before feeding it to the trained regressor. We use the same Feature Extractor and MLP for both scenes.



6.5.1 Neural Network Architecture

Feature Extractor The first part of the model is a *feature extractor* F , which receives as input a single depth image $I_{sc} \in I$ and outputs a feature vector $f_{sc} = F(I_{sc})$ that describes it. This feature extractor is composed of three different components, shown in Figure 6.6. First, the image is processed by a Convolutional Neural Network (CNN) that outputs a dense latent representation. We use a ResNet-18 [He+16] pre-trained on ImageNet [Den+09], which we finetune with our data. To reduce the gap between real and synthetic images, we equalize the image before feeding it to the feature extractor, during both training and evaluation. Then, the output of this CNN is passed through a *Self-Attention* [Zha+19b] module, which helps the model learn non-local dependencies, enlarging the model receptive field, so it accounts for distant information in the input images. Self-Attention mechanisms were originally designed for language models [Vas+17] but have recently demonstrated significant efficacy for computer vision tasks [Ram+19; Zha+19b; ZJK20; Han+22b]. We add a single Self-Attention layer, as they are expensive to train and evaluate.

Finally, we perform pooling operations to transform the output of the Self-Attention layer to a feature vector, f_{sc} , of a fixed size. In addition to the commonly used max-pooling [SZ15], we further concatenate it with the output of average-pooling, which has been shown to improve the performance of attention modules [Zho+16; HSS18; Woo+18]. The feature vector f_{sc} is thus

Table 6.1: Ablation study of the neural architecture and data augmentation. From left to right, we build upon our baseline and progressively add: simulation-space data augmentation, image-space data augmentation, pre-training, self-attention, and average-pooling. On both MAE (ℓ_1) and correlation (s) metrics, we observe increased performance on the validation set in every added component. Using a pre-trained network for feature extraction yields the largest gains. We use a color code to highlight **best** and **worst** cases.

Metric	Parameter	Baseline	Data Augmentation			Architecture		
			w/ sim	w/ image	w/ pre-Train	w/ attention	w/ pooling	
$\ell_1 \downarrow$	kStretchWeft	0.122	0.118	0.112	0.087	0.071	0.071	0.071
	kStretchWarp	0.109	0.101	0.102	0.074	0.066	0.066	0.061
	kStretchBias	0.050	0.043	0.046	0.043	0.042	0.042	0.038
	Avg. Stretch	0.094	0.087	0.087	0.068	0.060	0.060	0.057
	kBendingWeft	0.095	0.093	0.091	0.084	0.082	0.082	0.072
$r \uparrow$	kBendingWarp	0.124	0.119	0.112	0.110	0.101	0.101	0.094
	kBendingBias	0.086	0.081	0.086	0.081	0.068	0.068	0.060
	Avg. Bending	0.102	0.098	0.097	0.092	0.084	0.084	0.075
	kStretchWeft	0.445	0.475	0.453	0.643	0.788	0.788	0.798
	kStretchWarp	0.543	0.503	0.510	0.645	0.715	0.715	0.771
$s \uparrow$	kStretchBias	0.477	0.508	0.608	0.701	0.733	0.733	0.781
	Avg. Stretch	0.488	0.495	0.523	0.660	0.745	0.745	0.783
	kBendingWeft	0.611	0.684	0.781	0.783	0.798	0.798	0.863
	kBendingWarp	0.614	0.654	0.747	0.772	0.806	0.806	0.921
	kBendingBias	0.624	0.828	0.837	0.872	0.893	0.893	0.942
$s \uparrow$	Avg. Bending	0.616	0.722	0.788	0.809	0.832	0.832	0.909

a concatenation of max-pooled features and average-pooled features:

$$\mathbf{f}_{sc} = \left\{ \mathbf{f}_{sc}^{max} \oplus \mathbf{f}_{sc}^{avg} \right\}.$$

Fusion Our design allows us to combine features $\mathbf{f}_{sc}$ from more than one scene into a single feature vector $\mathbf{f}$. For every image in I , we compute $\mathbf{f}_{sc}$. We then fuse those feature vectors into a single vector by performing pooling across $\mathbf{f}$. Similarly to $\mathbf{f}_{sc}$, $\mathbf{f}$ is composed of features: $\mathbf{f} = \left\{ \max_{sc} \left\{ \mathbf{f}_{sc}^{max} \right\} \oplus \left\{ \text{avg}_{sc} \left\{ \mathbf{f}_{sc}^{avg} \right\} \right\} \right\}$. The max-pooled features are fused using the maximum value across scenes, while the average-pooled features are fused using the mean across scenes. We illustrate this procedure in Figure 6.6.

Parameter Regressor Our last component is a fully-connected Multi-Layer Perceptron (MLP), which takes as input the feature vector $\mathbf{f}$ and the material density, ρ , and outputs the simulation parameters $\hat{\mathbf{P}}$. As our loss function, we compare the real parameters $\mathbf{P}$ with the model estimations $M(I, \rho) = \hat{\mathbf{P}}$ using an ℓ_2 norm.

6.5.2 Quantitative Evaluation

In this section, we quantitatively evaluate the performance of the method for estimating the mechanical parameters. We validate the design choices of the model and evaluate our results depending on the type of input used.

Ablation Study of the Model Design

We aim to understand the effective contribution of the data augmentation strategy, and the network architecture design. For these experiments, we randomly split the synthetic data in 90% for training and 10% for validation, using the same split for every experiment. Results are shown in Table 6.1. As our baseline, we use a simple model without *self-attention*, without *average-pooling* features, where the CNN backbone is randomly initialized, and without data augmentation. From this baseline, we progressively add different components and measure the performance of the validation data using Mean Absolute Error (MAE) and Spearman correlation (r). The parameters are normalized using the minimum and maximum values of the training set.

Given the training configuration with all the data augmentation –which provides a small increase in performance most likely because the validation dataset is synthetic data– we evaluate the neural architecture.

Using a CNN backbone pre-trained on ImageNet [Den+09] instead of a randomly initialized one, we observe a significant increase in model performance across every parameter and metric. Training a feature extractor that receives

images from both scenes at the same time would not allow us to leverage pre-training, which would negatively impact generalization. Then, we add a *self-attention* [Zha+19b] layer after the CNN backbone, which allows the model to integrate information that is present in distant areas of the images. Interestingly, this module significantly helps predict the *kStretch* parameters while having a more minor influence on the *kBending*. Finally, adding *average-pooling* in addition to the commonly used *max-pooling* has a highly positive impact on the error rates. We use this last configuration with all the components for all the results shown in the chapter. It is worth noting that the *kBendingBias* is easily predicted by the model, even in its most basic configuration. This is likely because this parameter correlates most strongly with the density of the material (shown in Figure 6.4), so the model can leverage this information for the predictions.

Evaluation of Input Influence

The design of our method supports taking as input one or multiple images. In this experiment, presented in Table 6.2, we evaluate the error testing different configurations of the input. Note that we train a different model for each configuration. In every case, using the *density* as input helps the model to generalize. This is particularly relevant for *kBending* parameters, for which the *density* alone provides more information than depth images. The *stretch* scene is typically more informative than the *bending* one, as both MAE and correlations are usually better when it is provided. When using both scenes simultaneously, the model provides more accurate estimations than any of the scenes individually, showing that the two scenes provide complementary information.

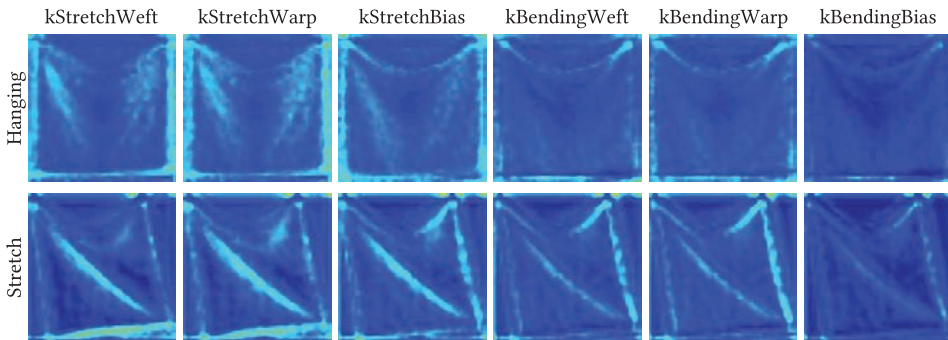
Table 6.2: Results for real depth images varying the input. From left to right, the input is: only density, only depth images, and both density and depth. The best results are obtained using every data source as input. For *kBending*, the *density* alone provides more information than only depth images. We use a color code to highlight **best** and **worst** cases.

Metric	Parameter	Density			OnlyDepth			Density & Depth		
					Stretch	Hanging	Both	Stretch	Hanging	Both
$\ell_1 \downarrow$	kStretchWeft	0.113	0.102	0.107	0.062	0.054	0.056	0.051	0.051	0.051
	kStretchWarp	0.091	0.067	0.081	0.056	0.055	0.059	0.052	0.052	0.052
	kStretchBias	0.034	0.039	0.054	0.034	0.033	0.036	0.031	0.031	0.031
	Mean Stretch	0.079	0.069	0.081	0.051	0.047	0.050	0.045	0.045	0.045
	kBendingWeft	0.142	0.233	0.213	0.145	0.128	0.139	0.125	0.125	0.125
$r \uparrow$	kBendingWarp	0.126	0.184	0.126	0.074	0.037	0.063	0.035	0.035	0.035
	kBendingBias	0.094	0.166	0.081	0.054	0.055	0.069	0.046	0.046	0.046
	Mean Bending	0.121	0.194	0.140	0.091	0.073	0.090	0.069	0.069	0.069
	kStretchWeft	0.184	0.407	0.403	0.418	0.712	0.469	0.728	0.728	0.728
	kStretchWarp	0.002	0.502	0.407	0.503	0.520	0.433	0.533	0.533	0.533
	kStretchBias	0.289	-0.141	-0.068	-0.007	0.367	0.383	0.550	0.550	0.550
	Mean Stretch	0.158	0.256	0.247	0.305	0.533	0.428	0.604	0.604	0.604
	kBendingWeft	0.483	0.052	0.267	0.625	0.673	0.683	0.717	0.717	0.717
	kBendingWarp	0.317	0.048	0.156	0.407	0.467	0.433	0.533	0.533	0.533
	kBendingBias	0.357	-0.044	0.108	0.250	0.333	0.417	0.546	0.546	0.546
	Mean Bending	0.386	0.019	0.177	0.427	0.491	0.511	0.599	0.599	0.599

Neural Saliency Maps

We aim to understand which parts of the scenes are most relevant for the model when making its predictions. To do so, we create saliency maps using *Axion-Based Class-Activation Mappings* [Fu+20], which averages the activations of the deep layers weighted by their importance with respect to each target parameter, and aggregate them over our real dataset. In Figure 6.7 we observe that each scene provides the model with different cues. For *kStretch* parameters, the model is sensitive to the central wrinkle of the *stretch* scene and the central fold of the *hanging*. For *kBending* parameters, the model is sensitive to areas in the borders of the fabric where small but noticeable wrinkles are present, in both scenes. As in other experiments, we observe that the model can find more relevant features on the *stretch* scene.

Figure 6.7: Saliency maps [Fu+20] aggregated per parameter. The model relies on the central areas of the fabric samples for predicting the stretch parameters. For bending, it is most sensitive to areas on the borders of the samples.



6.6 A Similarity Metric for Drape

The regressor introduced in Section 6.5 allows us to infer the mechanical parameters of a target fabric. However, since a direct interpretation of such parametric space is not human-friendly, it is very difficult to understand the residual errors shown in Tables 6.1 and 6.2. Do the regressed parameters produce a drape similar to the target image? Notice that, since the mechanical parameter space is non-orthogonal, small parameter changes may produce unexpected deformations. Therefore, we hypothesize that a *perceptual* similarity metric for drape is needed to interpret our quantitative results. We describe this metric next.

6.6.1 Image-based Similarity of Drape

Motivated by our hypothesis that the *hanging* and *stretch* scenes are sufficient to convey the fabric mechanics, we propose an image-based similarity

metric using renders of such scenes. Let $P \subset \mathbb{R}_7$ be the parameter space of our simulator, $P_a \in P$ and $P_b \in P$ two different parameter sets, and $a \sim R(P_a, \text{scene})$ and $b \sim R(P_b, \text{scene})$ two rendered simulations obtained for a certain scene configuration. We define a distance metric for a particular scene as:

$$d_{\text{scene}}(P_a, P_b) = \frac{\sum_{i=1}^N \sum_{j=1}^N \text{IM}(a_i, b_j)}{N^2} \quad (6.1)$$

where IM is an image-space distance metric, and N is the number of different simulations we run. Since real cloth is very sensitive to parameters such as initial state or initial shape, in order to learn a metric that is robust to real-world conditions, we perturb the initial state and boundary conditions in a set of simulations using random jittering to the initial forces. We empirically found that averaging over multiple simulations for the same set of (P_a, P_b) gives us a more informative metric.

Furthermore, we take into account both *hanging* and *stretch* scenes, hence our final metric is defined by averaging their distances across both scenarios, resulting in our final metric:

$$d(P_a, P_b) = \frac{d_{\text{hanging}}(P_a, P_b) + d_{\text{stretch}}(P_a, P_b)}{2} \quad (6.2)$$

Note that we propose a similarity metric, which, as opposed to real distance metrics, does not necessarily have to meet the metric axioms [TK74]: it can produce asymmetric values, violate the triangle inequality, and does not need to define what the identity is. According to Equation 6.1, the distance of one fabric with itself is not necessarily zero; it just needs to satisfy a minimum requirement where the distance of every material with itself should be smaller than the distance of any material with any other material,

$$d(P_a, P_a) < d(P_a, P_b), \forall P_a \neq P_b \quad (6.3)$$

In order to remove the possible influence of optical properties, scene illumination, and camera parameters, we use the same scene configuration for every render, with grayscale albedo and a Lambertian BRDF.

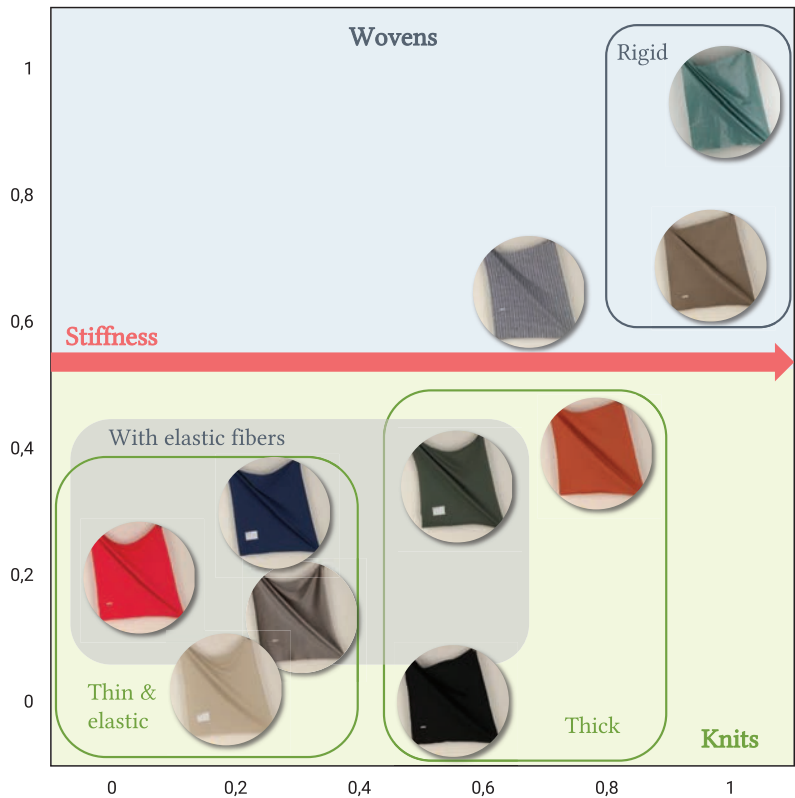
For IM we use LPIPS [Zha+18] which we empirically found to perform better than other alternative image metrics. We find that metrics based on pre-trained neural networks work better than lower-level alternatives, while content-aware distances are more powerful for this purpose than style-aware metrics. This suggests that the size, position, and shape of the wrinkles and deformations of the fabrics are important factors that explain differences between materials. We empirically found that $N = 5$ simulations are typically enough, as more samples provide very marginal improvements.

See the supplementary material of this chapter for more details about the proposed metric.

6.7 Evaluation

We propose to evaluate our method from Section 6.5 and the metric from Section 6.6 by comparing our estimations with human ratings (recall that we provide quantitative errors per parameter in Section 6.5.2). To this end, in Section 6.7.1, we first collect a large number of ground truth human judgments about the similarity of triplets of fabrics. Then, in Section 6.7.2, we demonstrate that our image-based metric using ground truth parameters, as well as the estimated ones, encode the same preferences. Finally, in Section 6.7.3, we show qualitative

Figure 6.8: Human Judgments 2D tSTE embedding [Tam+11] computed from human perceptual judgments about real fabric similarity (i.e., ground truth). We observe interesting patterns: wovens and knits are separated; elastic materials are clustered; thick and thin materials are separated. Neither axes directly correspond to any material property, instead they emerge from the embedding.



comparisons and demonstrate the usefulness of our approach in a downstream task consisting of ‘search by similarity’.

6.7.1 Human Judgment Perceptual Similarity of Drape

We then use ten samples from our real dataset with known ground truth mechanical parameters and setup the user study as follows. Participants are presented with a triplet of fabrics and, using one fabric as a reference, they are asked which of the two remaining fabrics is most similar [Zha+18; Gar+14] to the reference fabric. Participants are encouraged to manipulate the samples and focus only on the mechanical similarity and overall drape, and to ignore properties like material reflectance. Each participant rated 20 triplets that were pseudo-randomly sampled, ensuring that at least each of the ten test fabrics is used twice as a reference. Given the same triplet, we observe an average of 86.68% agreement between our participants across all experiments and materials, suggesting that there is a perceptual understanding of fabrics mechanics that humans share. We did not observe any significant differences in agreement depending on the volunteer demographics or level of expertise in fabric handling or simulation.

Leveraging the user study described above, we can compute an embedding that captures the relative distance between real materials according to human perception (i.e., ground truth perceptual similarity). Figure 6.8 depicts such embedding in 2D, computed using tSTE [Tam+11], which allows us to calculate perceptual distances between materials using the Euclidean norm. The embedding depicts many interesting patterns, including perfectly separated woven and knitted fabrics; elastic and thin fabrics are clustered together; thick and thin knits are well separated. These patterns suggest that fabric structure, composition, and density play an important, non-linear role in the overall perception of the mechanical properties of fabrics.

6.7.2 Image-based vs. Human Perceptual Similarity

We evaluate the agreement between humans and our estimations within the context of a similarity rank. For each fabric of our real dataset, we compute the distance to the rest of the materials using different metrics: 1) the Euclidean distances on the Human Judgments tSTE embedding shown Section 6.7.1; 2) the z-score distances in the parametric space of the mechanical simulation, P ; 3) the distance using our similarity metric for drape explained in Section 6.6. We compare ground truth parameters and estimated ones for the second and third cases. The summary of results is shown in Table 6.3, and the complete analysis is presented in the supplementary material.

First, we demonstrate that our similarity metric using ground truth parameters correlates with human judgments. Figure 6.9 (top) illustrates the outcome using Spearman correlation(r). We observe that the correlation for each fabric is higher than 0.8, with an average of 0.893, showcasing a strong correlation. These results suggest that our metric, which only takes images of the *hanging* and *stretch* scenes as input, can measure distances between materials as humans would. Then, as shown in Figure 6.9 (bottom), we use our estimated parameters instead of the ground truth, reaching an average correlation of 0.680. Even though this value is slightly smaller than the ground truth, it is still significant to conclude the estimations of our model agree with human judgments. Note that the materials with higher correlation are those lying on the extreme areas of the embedding obtained in Figure 6.8 that have very distinct characteristics. Likewise, we also compute the distance for each material using merely the parameter spaces of the simulation. As can be seen, using this space does not produce correlated outputs with human judgments, reaching correlations below 0.44 in any case tested.

Figure 6.9: Correlation between the ordering provided by the Human Judgments (x-axis) and our drape similarity metric with (top) the Ground Truth simulation parameters (y-axis), and (bottom) the estimations of our model. We plot z-scores instead of the raw distances to help visualization.

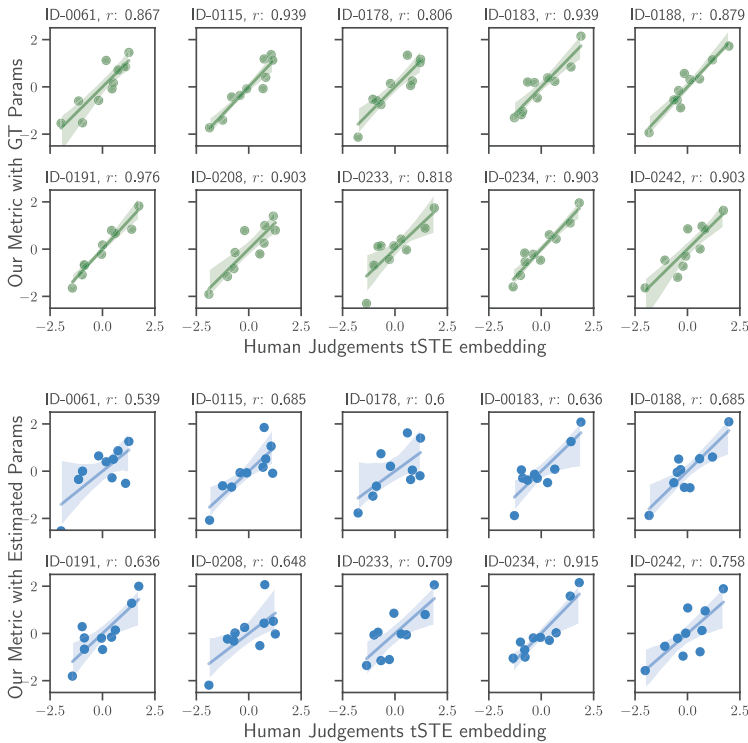


Table 6.3: Average ($\pm$ std.) correlation with rankings obtained through the human judgments tSTE Embedding, depending on the parameter source (ground truth or predicted), and metric used to compute similarity (parameter distance or our drape similarity).

	Parameter Distance	Drape Similarity Metric
GT	0.431 $\pm$ 0.17	0.893 $\pm$ 0.05
Estimated	0.377 $\pm$ 0.19	0.680 $\pm$ 0.09

6.7.3 Qualitative Results

Figure 6.10 compares the simulations obtained with the parameters of our method with the ground truth parameters for a few examples. We can observe that our estimations are very close to the ground truth in this cases. Finally, our metric can be used to search between materials of similar drape. We illustrate this in Figure 6.11. As shown, a naive ranking using directly the parameter space does not provide any meaningful ordering. On the contrary, using our similarity metric, we obtain ranks that agree with those given by the human embedding, showcasing the potential of our automatic metric to explore fabric collections.

Limitations Even if our model provides accurate predictions, its estimations are not always truthful to the real materials. We illustrate this in Figure 6.12, where the model predicts fewer bends on the final drape than what the ground truth generates. According to our metric, this prediction is closer to thicker materials than to what humans perceive as most similar to the reference material.

6.8 Conclusions

In this chapter, we have presented a casual method to estimate the mechanical parameters of fabrics from depth images of fabric samples placed at two specific configurations. We have validated our architecture and inputs numerically, proving that all the components of our method are necessary to provide accurate estimations. While our quantitative analysis helped us understand the importance of each component, we found that these errors are not interpretable, nor do they help us understand the overall appearance of the predicted drape. Therefore, we have presented the first metric, which, by purely working on the image space, can capture differences in fabric mechanics as humans do. We have used such metric for two purposes: first, to validate the accuracy of our estimated parameters perceptually, and second, to showcase a novel application of search by drape similarity.

Figure 6.10: A comparison between the simulations obtained through the ground truth parameters, and those obtained using the predictions of our model, from a representative set of fabrics of our test set. As shown, the estimations of the model yield similar drapes to those of their ground truth counterparts, which we also evaluate quantitatively.

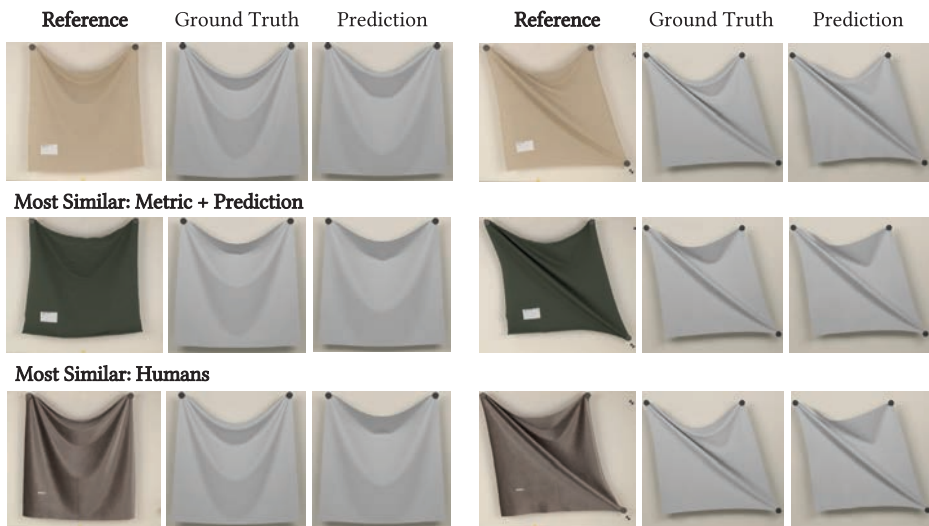


Our work could be improved in several ways. Our neural network is trained using a pure regression loss. Training the network using differentiable simulation could improve training, and help generalization and error interpretation. We could incorporate our perceptual metric as a loss function. However, it requires multiple differentiable simulations and a deep feature extractor, which will result in a significant computational overhead. In addition, less expressive simulation engines may correlate less with human perception. Similarly, we would like to scale our training dataset and user study to handle more and more diverse samples, to cover a broader variety of fabric families. Interesting possible extensions would include taking as input RGB images instead of depth maps, training with real samples, incorporating symmetry consistency losses, or learning a similarity metric that can work by using captured images as input (instead of simulations). Finally, we hope our work might inspire future work in the long-standing problem of validating fabric mechanics in a way that is agnostic to the simulator parameters.

Figure 6.11: Search by drape similarity. The ordering provided by the parameter space (first row) does not match human judgments (second row), while the arrangement obtained by our metric matches humans with high consensus (third row).



Figure 6.12: A failure case of our method. Given the reference material (first row) as input, the model predicts fewer bends than the ground truth. According to our metric, this prediction is closer to a thicker material (middle row), than to what humans perceive as most similar to the reference fabric (bottom row).



6.A Additional Implementation Details

Training. We train our model using PyTorch [Pas+19] as our learning framework and Kornia [Rib+20] for image-processing operations, including normalization and data augmentation. The model is optimized using AdamW [LH17] for a total of 50 epochs, with a learning rate of $\alpha = 0.0003$, which is halved every 10 iterations. All other optimizer hyperparameters followed the default AdamW configuration. We use mixed precision training [Mic+18], a batch size of 256, and a weight decay of 0.000002. The input images are of 180×180 pixels, and we use a Mean Squared Error (MSE, ℓ_2) as our loss function for our regression problem. Model size design and hyperparameter selection were conducted using Bayesian hyperparameter tuning [Bie20]. Training takes approximately one hour on an NVIDIA 2080 RTX GPU.

Network Design. Our feature extractor is a pre-trained ResNet-18 [He+16] (TorchVision pretrained weights checkpoint weights='IMAGENET1K V1'), followed by a Self-Attention layer [Zha+19b] with 64 attention heads. The output of this layer is max and average-pooled, and a final multi-layer perceptron (MLP) with 4 hidden layers with 512 hidden units each, receives the pooled features and the density value, and outputs the 6 predicted mechanical parameters. The MLP layers use ReLU [MHN13] non-linearities, followed by Layer Normalization [BKH16], and a Dropout [Sri+14] rate of 0.5. Both the feature extractor and the MLP are trained simultaneously.

Data Augmentation. We use the following policy, in order: First, we randomly rescale the input images using (0.5, 1.5) rescale ranges. We then randomly erase [Zho+20] parts of the input images ($p = 0.2$), randomly rotate them ($p = 0.2$, $angle = \pm 10^\circ$), apply a random perspective change ($p = 0.5$, $scale = 0.5$), and a random thin plate spline warp ($p = 0.2$, $scale = 0.5$). We then crop the center area, with a resolution of 180×180 pixels. To those images, we randomly change their contrast and brightness, on the (0.5, 1.5) ranges. We then apply a final set of transformations, including image equalization, random horizontal flip ($p = 0.5$), random Gaussian noise ($p = 0.5$, $\mu = 0$, $\sigma = 0.02$), random posterization ($p = 0.5$), random sharpening ($p = 0.5$), and random Gaussian blur ($p = 0.5$, kernel size of 3×3). Please refer to the Kornia documentation [Rib+20] for specific implementation details of each of these transformations and the specific meaning of the mentioned parameters.

6.B Additional Results

In this section, we show the relative similarity rankings computed through the tSTE embedding of our user study, compared to the ground truth and predicted parameter distances, as well as the drape similarity computed using the ground truth and predicted parameters. We also provide additional details of our user study.

Figure 6.13: Relationship between the distance obtained by our user study, our distance metric, and the difference in each parameter. On the top row, we plot the difference between the value of a parameter against the average distance between fabrics we obtained through our user study. On the bottom, we show the same parameter distances, now against the distance predicted through our perceptual metric. As shown, no mechanical parameter dominates the distance predicted by either the user study or our metric. The perception of mechanical properties can be understood as a highly non-linear phenomenon in which many parameters interact in complex ways. We use z-scores to help visualization.

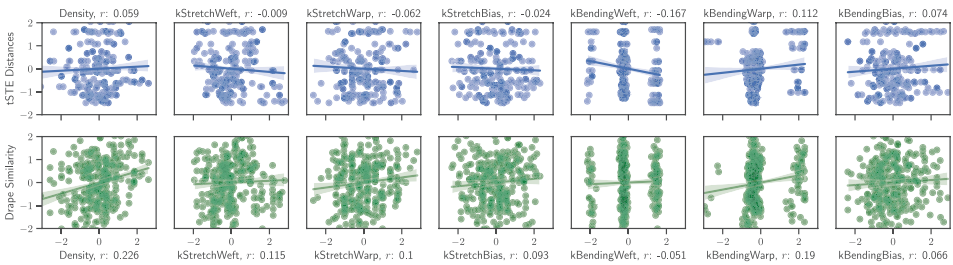
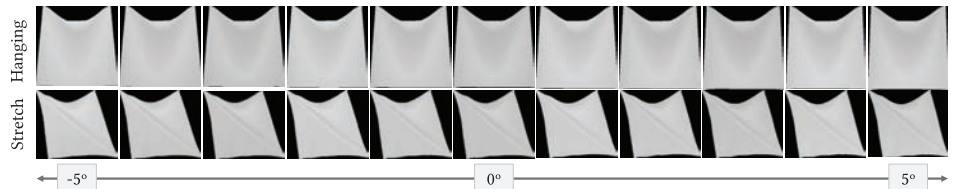


Figure 6.14: To recreate possible misalignments that may be present on the real images, we propose a *simulation-space* data augmentation policy, in which we simulate the drapes from different rest positions, on the $[-5, 5]$ degree ranges. For every scene, we generate 11 different simulations uniformly in this range, effectively creating a dataset of images for a single material.

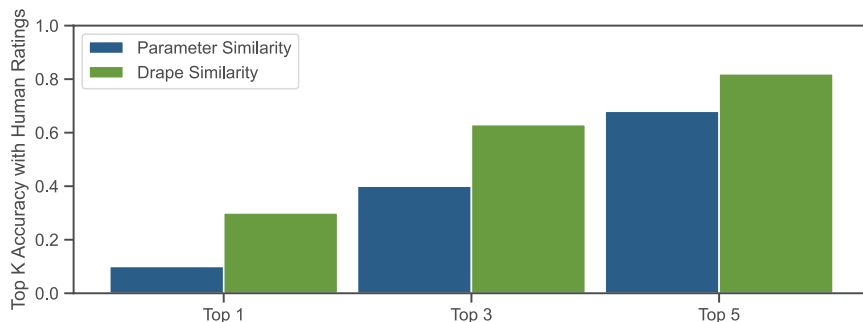


User Study: Procedure and Stimuli

Stimuli. We use ten physical fabric samples of our real dataset, which are representative of different types of fabric structures, thicknesses, densities,

and mechanical properties. We leverage 25×25 cm samples, which is an adequate size for the manipulation of the materials.

Figure 6.15: Ranking classification accuracy on different ways of comparing materials. Comparing materials simply by measuring their distances on parameter space yields inaccurate similarity rankings, as the parameter space is not orthogonal and cannot be linearly correlated with human perception of material similarity. Our Drape similarity metric provides us with a way of assessing the similarity between materials on a more perceptual-aware space, yielding more accurate rankings.



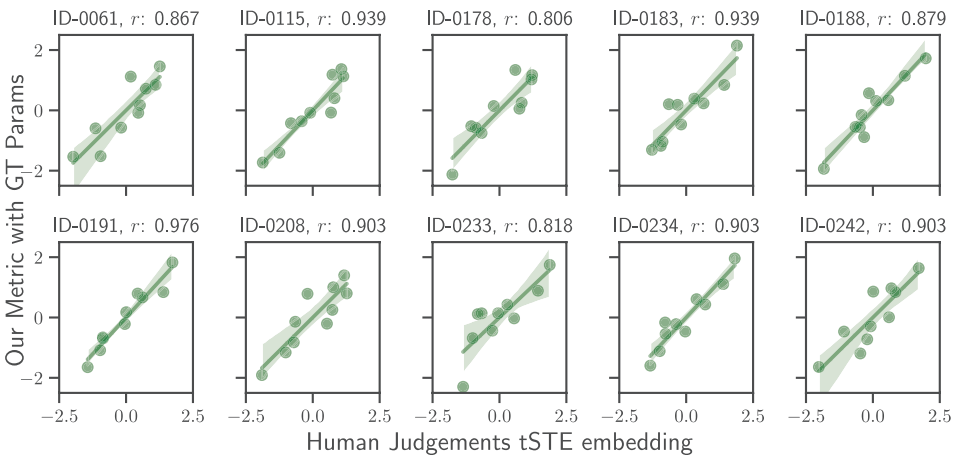
Participants. A total of 30 volunteers took part in the test. The participants had different backgrounds and diverse levels of expertise in physical manipulation of textiles. A study on their demographics can be found in Figure 6.20.

Procedure. Following previous work on human perception [Zha+18; Gar+14], we design a Two-Alternative Forced Choice (2AFC) user study in which participants are presented with triplets of physical fabric samples. We performed the experiment in a lab-controlled environment because the fabric samples had to be physically manipulated. Participants were asked to select which of two fabrics is most similar to a reference fabric. They were suggested to manipulate the samples (stretching, bending them, etc.) and to focus only on mechanical similarity, thus ignoring other factors such as optical properties (color, specularity, transparency) or irrelevant tactile feedback (softness, for instance). Material reflectance was thus ignored by participants. Volunteers were not instructed to leverage fabric motion to make their decisions but were free to dynamically manipulate the samples, which some participants did. Following recommendations in user study design for graphics [Byl+22; Dod+22], we informed the users that there are no right or wrong answers, allowed them to use their own criteria for answering the questions, and self-identify their demographics and levels of expertise. The volunteer demographics can be found on 6.20.

Each participant rated 20 triplets that were pseudo-randomly sampled following some rules: each of the 10 fabrics in our test set is shown as a reference at least twice to each participant, the order of the triplets is random, and each triplet was evaluated by at least 2 participants. Each test took between 15 to 30 minutes, depending on the participant, and was followed by an open discussion, to better understand which criteria each participant used to make their decisions. We started each test with a brief interview to register the self-reported participants' demographics [Dod+22], we continued it with the 20 perceptual comparisons and finished it with an open discussion so as to better understand which factors each participant used to make their decisions.

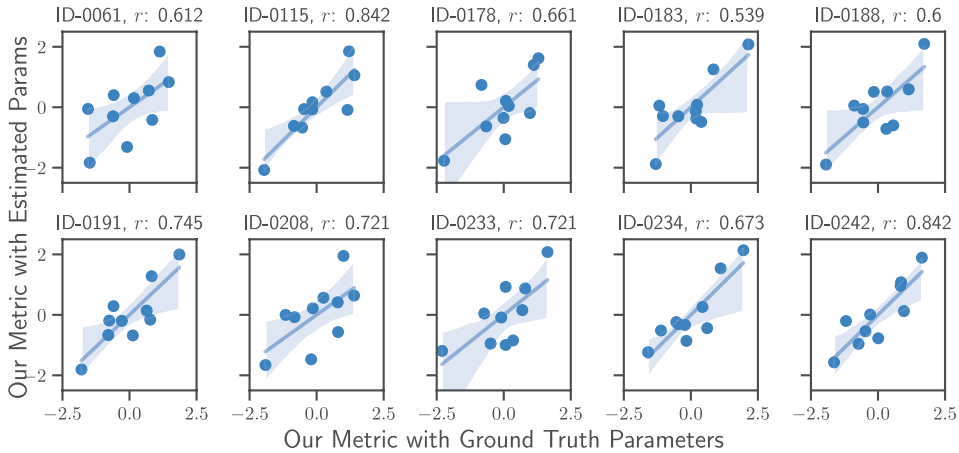
Results: Agreement 30 participants, with diverse levels of expertise with fabrics and computer simulations, volunteered for this user study. Given the same triplet, we observe an average of 86.68% agreement between our participants across all experiments and materials, suggesting that there is a perceptual understanding of fabrics mechanics that humans share. We did not observe any significant differences in agreement depending on the volunteer demographics, or level of expertise on fabric handling or simulation. This suggests that the perception of similarity of the mechanical properties of textiles may be relatively universal. When prompted, participants generally admitted that some triplets were easier to evaluate than others, either because neither candidate fabrics were similar to the reference or because both were very similar and thus difficult to make a definitive decision. Participants tended to evaluate elasticity and the shape of the fabric

Figure 6.16: Correlation between the ordering provided by the Human Judgments (x-axis) and our Drape Similarity Metric (y-axis). We plot z-scores instead of the raw distances to help visualization.



when it bends. When in doubt about those factors, a standard procedure was to assess similarity using the perceived weight of the material.

Figure 6.17: Correlation between the ordering provided by our similarity metric with the Ground Truth parameters (x-axis) and the estimated ones (y-axis). We plot z-scores instead of the raw distances to help visualization.



6.C Ablation Study: Similarity Metric Parameters

Here, we study the factors that might impact the performance of the metric: the image-based similarity metric (IM), the number of simulations necessary to account for the non-determinism of the simulation (N), and the scene (*hanging* or *stretch*).

- **Choice of IM** The image-based metric should be able to capture the subtle differences in wrinkles, folds, and overall shape of the simulated drape. While previous work has learned this metric from data [Gar+14; Lag+19], we proposed a simpler approach using existing image-based similarity metrics. Figure 6.21 (a) shows results for several deep and non-deep methods. For deep learning-based metrics, it is particularly relevant that the metric can account for local and positional differences. A *Content loss* or the *LPIPS* shows stronger performance than translation-invariant losses (e.g. *Style Loss* or *DISTS*).

- **Choice of N** We study how many simulations that are needed to account for the non-determinism of the simulation. Figure 6.21 (b) shows that $N \geq 5$ is enough to obtain accurate results.
- **Choice of scene** We study which scene is best for estimating drape similarity in Figure 6.21 (c). The *stretch* scene generally provides more information than the *hanging* scene, while using both achieves the best agreement overall.

Figure 6.18: Correlation between the ordering provided by the tSTE embedding from our user study (x-axis) and our distance metric, computed using the estimated parameters (y-axis). We plot z-scores instead of the raw distances to help visualization.

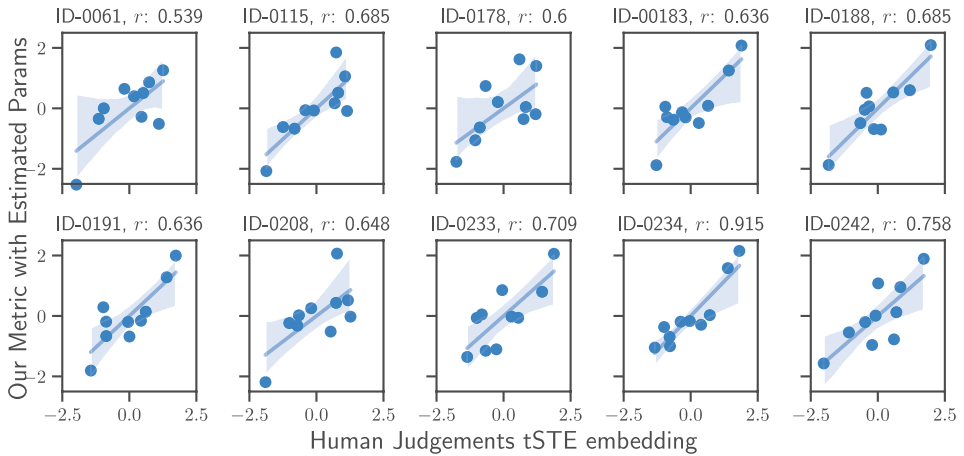


Figure 6.19: Correlation between the ordering provided by the tSTE embedding from our user study (x-axis) and the distance computed on the parameter space, for the model predictions (y-axis). We plot z-scores instead of the raw distances to help visualization.

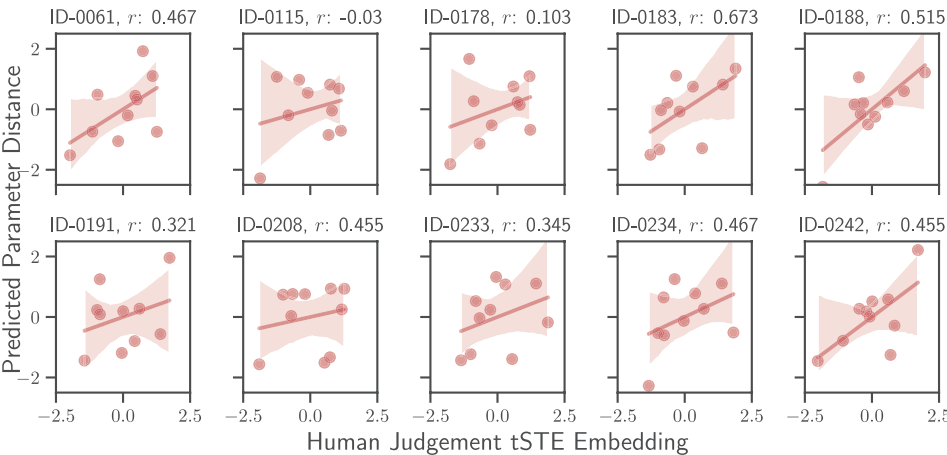


Figure 6.20: Demographics of the participants of our perceptual study. Following recommendations on user studies for computer graphics [Dod+22; Byl+22], we capture the self-identified gender, age, and education level of each participant. We also asked them to identify their level of expertise with handling physical textile materials, as well as their expertise in computer simulation and rendering of fabrics.

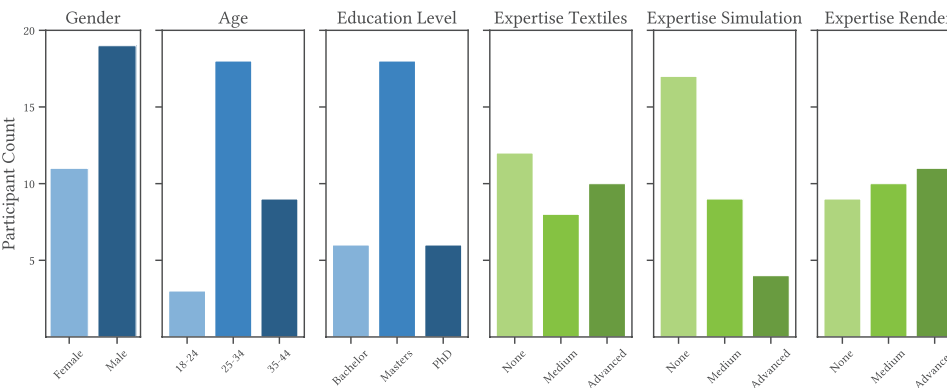
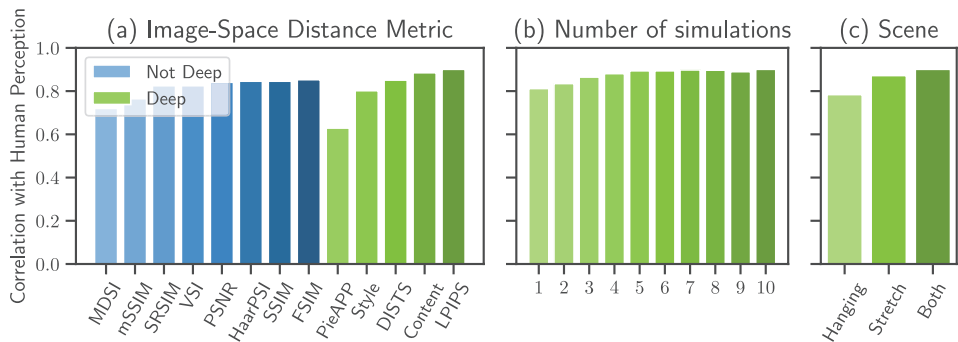
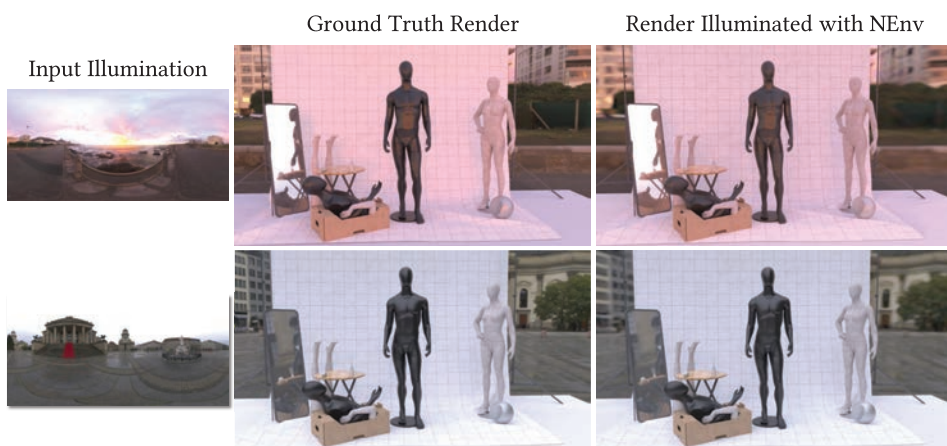


Figure 6.21: Ablation study on different factors of our similarity metric. Y-axis is Spearman correlation. (a) Results for several image-based metrics IM (we use the implementations provided in *PIQ* [KZP19]). (b) Number of simulations needed to account for the non-determinism (optimal choice is 5). (c) Impact of the scene.



CHAPTER 7. NEURAL MODELS FOR GLOBAL ILLUMINATION

Figure 7.1: Renders using analytical illumination methods with multiple importance sampling and our learned lighting approach for two environment maps.

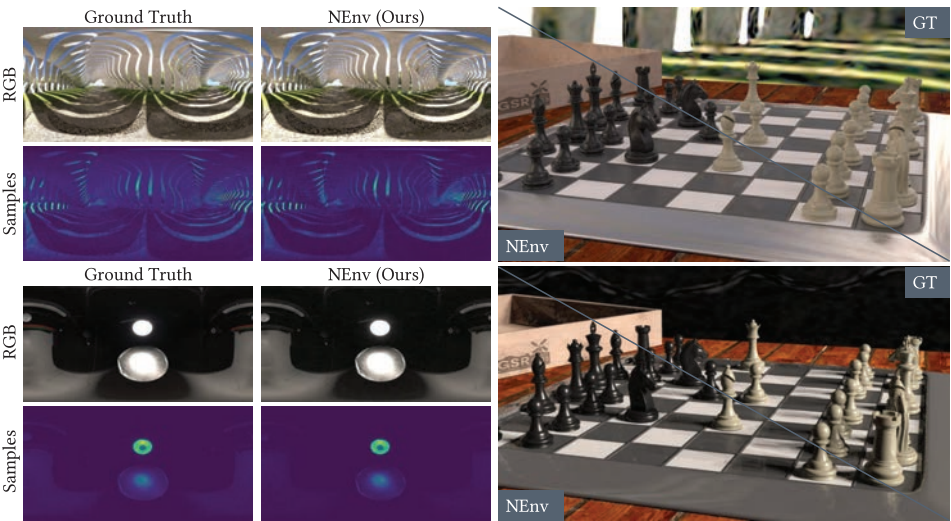


Environment maps are commonly used to represent and compute far-field illumination in virtual scenes. However, they are expensive to evaluate and sample from, limiting their applicability to real-time rendering. Previous methods have focused on compression through spherical-domain approximations, or on learning priors for natural, day-light illumination. These hinder both accuracy and generality, and do not provide the probability information required for importance-sampling Monte Carlo integration. In this chapter, we propose NEnv, a deep-learning fully-differentiable method, capable of compressing and learning to sample from a single environment map. NEnv is composed of two different neural networks: A normalizing flow, able to map samples from uniform distributions to directions of the environment map, also providing their corresponding probabilities; and an implicit neural representation which compresses the environment map into an efficient differentiable function. The computation time of environment samples with NEnv is two orders of magnitude less than with traditional methods. NEnv makes no assumptions regarding environment map content, achieving higher generality than previous learning-based approaches. The contributions in this chapter led to the following publication, currently under review:



"NEnv: Neural Environment Maps for Global Illumination"
Carlos Rodriguez-Pardo, Javier Fabre, Elena Garces,
Jorge Lopez-Moreno
Under Review (2023)

Figure 7.2: From left to right, Ground Truth and our learned RGB environment map compression and sampling algorithms, and renders using both illumination configurations on a complex scene.



7.1 Introduction

Environment Maps are widely used in rendering to represent far-field illumination around a point with a single texture, usually a High Dynamic Range image (HDRI). There are two typical scenarios for using these maps: as infinite spherical light sources in offline rendering methods (such as path tracing), or as local probes that encode near-field irradiance to approximate global illumination in real-time applications.

In the first case, many algorithms such as Multiple Importance Sampling (MIS) [VG95] require the ability to sample lights in the scene, which can be problematic when dealing with environment maps. Since the source data comes from an image, there is no analytic Cumulative Distribution Function (CDF) that can be used for sampling. Instead, tabulated methods can be used, but these require a pre-computation step that consumes a considerable amount of memory, as well as a search step to find the sample probability.

In real-time scenarios, irradiance sampling is becoming more relevant with the increasing capabilities of GPUs and sample reservoir techniques for global illumination [Bit+20; Ouy+21], yet the usual techniques to sample environment maps do not easily benefit from parallel GPU environments. Also, in real-time, analytical approximations (such as Spherical Harmonics or Spherical Gaussians) are often used to overcome this issue, however, they are not able to capture final-scale details present in most environment maps.

This work presents a novel learning-based method for sampling, PDF evaluation, and compression of high-resolution environment maps, able to encode any type of illumination with high fidelity. Our method builds on recent advances in neural representations, that we found particularly suitable for this kind of problem, where it is important to encode both the high-frequency and low-frequency patterns. Key to our solution is to use a normalizing flow [Dur+19a] to encode an *invertible* representation of the environment map PDF, reducing both sampling and evaluation time by two orders of magnitude compared with analytic solutions. Thanks to being invertible, our representation is differentiable and compatible with importance-sampling techniques. We also propose a method based on implicit neural representations to compress the environment map, reducing the memory footprint of the original image with minimal loss. We demonstrate that our approach works for a wide range of scenes and is faster and more accurate than previous methods.

7.2 Related Work

In offline rendering, environment maps are usually encoded in high resolution ($> 4K$) HDR images (Radiance HDR [LS98], OpenEXR [KBH03]), requiring tabulated approximations for sampling and evaluation. Techniques such as Piecewise-Constant 2D Distributions [PJH16] or Hierarchical Warping [Cla+21] are used for this purpose, but they are not suited for RT applications due to limited time and memory budget, being a costly and difficult-to-parallelize operation, even for offline rendering. This cost can be reduced, for instance, by taking into account scene information, such as occluders [Aga+03] to avoid poor-quality samples.

Heitz [Hei20] proposed a method to invert non-analytically invertible CDFs, which can be applied to some distribution functions (e.g.: BSDFs) to obtain an analytical sampling function. However, environment maps are not good candidates for triangle-cut parametrization, since their PDF and CDF come from a discrete tabulated source (HDR image).

To avoid using high resolution images, real time methods rely on environment map prefiltering [Kau+00], or analytical approximations, such as Spherical Harmonics (SH) [See66], and Spherical Gaussians (SG) [Xu+13]. These methods allow for analytical evaluation of an environment map, as well as analytical sampling (directly, in the case of SG, or using Hierarchical Sampling for SH [JCJ09]) and easy interpolation (useful when not enough samples can be requested). Still, they do not accurately represent complex HDR images commonly used when representing high-detailed light setups by using environment maps, and blending important light features in the original images, sometimes generating artifacts. More recently, neural representations have been explored for this purpose. For instance, Gardner *et al.* [GES22] propose a conditional neural field representation, based on a variational auto-decoder (RENI), which leverages natural image priors to efficiently encode a full HDR environment image into a few-dimensional latent space vector. Being low-dimensional, it shares the same accuracy limitations that its predecessors, and does not consider sampling, or evaluation in its design. In the following, we describe the most relevant neural approaches for sampling and 2D image representations.

7.2.1 Neural Sampling and Representations

Learned Sampling and Normalizing Flows Normalizing Flows (NFs) have been proposed as powerful models for learned sampling for rendering applications. The seminal work of Müller *et al.* [Mül+19] proposed *Neural Importance Sampling*, using *Piecewise-Linear* and *Piecewise-Quadratic Coupling Flows* for learned sampling for Monte Carlo rendering. This method showed the first successful use of these neural models for Path-Tracing, but their results showed a prohibitive overhead in runtime cost over previous methods. In a later work, Müller *et al.* proposed *Neural Control Variates* [Mül+20], an *Autorregressive Flow* which allows for more efficient unbiased integration. Beyond Monte Carlo integration, Flows have been used for BRDF representations in inverse rendering [CNN22]. Relatedly, Sztrajman *et al.* [Szt+21] propose a method for neural sampling of BRDFs. However, instead of relying on NFs, they propose an encoder network that maps from *Neural BRDF* to fitted Blinn-Phong parameters for which importance sampling is known. Besides these tasks, NFs have been utilized in computer graphics and vision for image [KD18; WZY22] and video [Kum+19] generation, compression [Hel+20], super-resolution [Lug+20], domain translation [Gro+20], and uncertainty quantification [Wan+22c; Sel+20]. We build upon *Neural Importance Sampling*

[Mül+19] and propose a lightweight model for sampling and PDF evaluation of environment maps, as we describe in Section 7.4. We show that the proposed architecture and training produce a single network that can be integrated into any rendering pipeline, providing a significant speedup in image-based lighting with importance sampling.

Implicit Neural Representations (INRs) have emerged in recent years as a powerful alternative to traditional representations for natural signals. They allow for differentiable, continuous, and expressive functions which can represent many types of data, such as images [Sit+20; Meh+21], video [Gao+21], or *Neural Fields* for rendering [Mil+20]. INRs typically build upon standard Multi Layer Perceptrons (MLPs) but benefit from working on the frequency domain to better handle higher frequencies, which are present on natural signals. These can be introduced to the models using fixed *Positional Encodings* [Mil+20; Bar+21], *Fourier Features* [Tan+20; Kar+21] or *Sinusoidal Activations* [GES22; Meh+21; Sit+20], among other approaches. These general-purpose models have been utilized extensively in inverse and neural rendering, for representing geometry [Tak+21; Yan+21], illumination [GES22; GMX22; Att+22], material reflectance [Baa+22; Fan+22; Szt+21; Kuz+22; Kuz+21; Yao+22] or entire scenes [Mil+20; Fri+22; Tan+22; Ver+22; Sit+21]. We refer the reader to the seminal surveys in *Neural Fields* [Xie+22; Tew+22] for more comprehensive reviews of these representations for neural rendering and computer vision. Besides their effectiveness for rendering, INRs have also been used for image [Str+22] and video [Rho+22] compression. We build upon these findings to design an efficient yet expressive INR for environment maps, as described in Section 7.5.

Figure 7.3: Overview of *NEnv*. We propose an invertible generative neural network $\mathcal{F}$ (i.e. a *normalizing flow*) which simultaneously learns to sample directions (θ, ϕ) from an environment map, as well as evaluating the probability of a given direction.

This model, illustrated on the left, can be integrated seamlessly into path tracer render engines, allowing for efficient multiple importance sampling. We additionally use an implicit neural representation $\mathcal{C}$ (second column) which learns to map from directions to linear RGB values, allowing for faster evaluation.

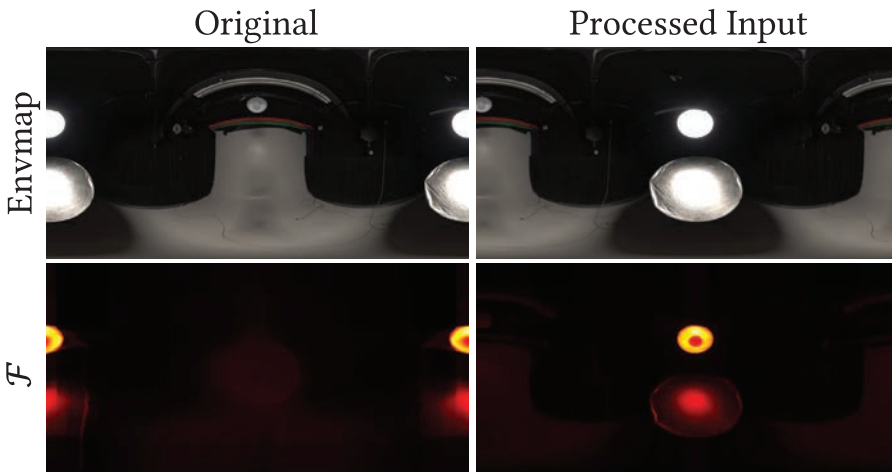


7.3 Overview

Our method takes as input a single HDRi environment map of any resolution, which is an image-based 360° representation of the illumination of a scene $I \in \mathbb{R}^{H \times W \times 3}$. First, we perform a series of pre-processing steps (Section 7.3.1) and compute the ground truth probability distribution function, (Section 7.3.2). Then, we train a normalizing flow F for joint sampling and PDF evaluation (Section 7.4), and an implicit neural representation c for environment map compression (Section 7.5). We illustrate NEnv in Figure 7.3.

F is an invertible neural network which can generate directional samples $(\theta, \phi) = F(z)$ by mapping from a known and simple noise distribution $z \in \mathbb{R}^2, z \sim p_z(z)$, as well as evaluating the probability density of any direction $p(\theta, \phi)$. F provides an efficient and differentiable approximation of the PDF encoded in I , which is particularly useful for Monte Carlo integration in rendering systems using Multiple Importance Sampling (MIS). We model c as a sinusoidal Multilayer Perceptron (MLP) which maps directions (θ, ϕ) to RGB values: $RGB = C(\theta, \phi)$; compressing I into a differentiable, continuous function. We train F and c individually for each input I , and design their architectures and training configurations to achieve high compression rates, efficient evaluation, and minimal loss in rendering quality. Once trained, both F and C can be integrated seamlessly into rendering engines, considerably reducing the time and memory consumption per sample compared to traditional global illumination approaches.

Figure 7.4: Environment map and the PDF encoded by NEnv for an unprocessed (left) and rotated (right) environment map. Our simple rotation preprocessing avoids discontinuities in the borders of the learned reconstructions, enhancing the performance of our models.



Implementation details are included in Section 7.6. To validate our method, we perform extensive ablation studies, and perform comparisons with previous work on environment map approximations (Section 7.8), both in environment map reconstruction and its impact on final renders.

7.3.1 Map Pre-processing

Some environment maps have large and intense light sources close to the horizontal borders of the image, separating the same light source into extreme values of $\phi \approx 0$ and $\phi \approx 2\pi$. While this is not a problem when rendering with traditional approaches, we observe these cases become very challenging for our normalizing flows, creating artifacts and tails in the encoded PDFs, as we show in Figure 7.4. We solve this by rotating the environment map by $\pi - \phi_{\max}$ angles, so that its most intense light source is in $\phi = \pi$, thus avoiding strong discontinuities at the borders:

$$\phi_{\max} = \operatorname{argmax}_{\phi} \sum_{\theta} \mathcal{I}(\theta, \phi) \quad (7.1)$$

While there are specific parameterizations for normalizing flows on spheres [Rez+20], we empirically observe that this rotation algorithm effectively solves the discontinuity issues we observed on real-world environment maps without requiring any modifications in our model design.

7.3.2 Computing Ground Truth Tabular PDFs

As mentioned, there are two widely-used approaches for sampling directions according to an environment map distribution:

- Computing a *Piecewise-Constant* [PJH16] 2D Distribution by sampling a marginal PDF to select a row/column of I , then sampling a conditional PDF to choose an individual pixel.
- A Hierarchical Warping [Cla+21; Pha19] algorithm based on MIP-mapping over I to iteratively warp 2D samples from an uniform space until they match the target distribution.

Pharr [Pha19] compared these methods, stating that despite obtaining different spatial warpings, both produce similar error in renders. Interestingly, the official implementation for Mitsuba3 [Jak+22] relies on the latter approach, while the available implementation for the 4th edition of PBRT [PJH20] uses the former. These are both widely-used engines in the literature. Due to its comparably better efficiency (see Section 7.8.3), which is important during model training, we choose to use the Piecewise-Constant method to compute our ground truth PDFs and samples.

To obtain the PDF for a given environment map I , we first define $f(\theta, \phi)$, by mapping directions to the luminance values stored in the image I . These values are proportional to the probability of selecting a particular pixel that we want to compute, but require additional processing and normalization. We start by multiplying I by $\sin(\theta)$, given the θ corresponding to each row, thus eliminating the distortion caused by mapping the image to the unit sphere [PJH16]. Using $f(\theta, \phi)$ as source data, we then compute one conditional density distribution $f_{\text{conditional}_\theta}(\phi)$ for each ϕ , so we can later obtain a single marginal distribution by integrating all conditional distributions (rows) to build 1D Piecewise distributions. We use this marginal distribution to sample a row, and in turn, sample the corresponding conditional distribution stored for such row to select a column, hence getting a single pixel defined by θ, ϕ coordinates.

Finally, to compute the probability density of choosing a given pixel $p_i(\theta, \phi)$, we obtain the CDF for the conditional distribution at row ϕ for value θ , and normalize it to the marginal distribution integral:

$$p_{\mathcal{I}}(\theta, \phi) = \frac{f_{\text{conditional}_\theta}(\phi)}{\sum_{i=0}^{n_\theta} f_{\text{marginal}}(i)} \quad (7.2)$$

where $f_{\text{conditional}_\theta}$ represents the data used to compute the conditional PDF for row ϕ , and f_{marginal} is the data used to compute the marginal distribution over the n_θ rows of the Y axis.

7.4 Learned Sampling and PDF evaluation

We aim to find a learning-based representation useful to sample directions from an environment map following its distribution, as well as evaluating the probability density (PDF) for any given direction $p(\theta, \phi)$. While this could be approached using two independent neural networks, there is no guarantee that a sample generated by a network is drawn with the probability estimated by the other network, surely affecting the illumination integral and producing visual artifacts (e.g. fireflies). Neural networks by default are typically non-invertible and can become prohibitively expensive once model sizes grow, either by stacking more layers or adding more neurons. We instead leverage *normalizing flows*, defined by specifying a bijective function F or its inverse F^{-1} . This enables both sampling and evaluation simultaneously and coherently with a single invertible neural network.

7.4.1 Formalization

We define our normalizing flow as a differentiable transformation F which maps from a random vector $z \in \mathbb{R}^2$, sampled from a 2-dimensional uniform distribution $p_z(z)$, to samples $(\theta, \phi), \theta \in [0, \pi), \phi \in [0, 2\pi)$:

$$(\theta, \phi) = \mathcal{F}(z) \text{ where } z \sim p_z(z) \quad (7.3)$$

The probability density of a direction $p_F(\theta, \phi) \in \mathbb{R}$ under the normalizing flow F is obtained through a change of variables. This can be computed analytically because F is designed as an invertible function. Formally:

$$p_F(\theta, \phi) = p_z(\mathcal{F}^{-1}(\theta, \phi)) \left| \det \frac{\delta \mathcal{F}^{-1}}{\delta(\theta, \phi)} \right| \quad (7.3)$$

F locally transforms the density of original noise distribution $p_z(z)$. The intensity of this change is measured by the determinant of the Jacobian of the transformation, as in the right hand side of the equation above. $p_z(z)$ is chosen to be a bidimensional uniform distribution bounded between 0 and 1: $p_z(z) = \mathbf{V}^2(0, 1)$.

As shown, F provides the two calculations required in MIS for Monte Carlo integration. However, F needs to meet two competing requirements: it has to be as accurate as possible, *i.e.*, close to the real PDF, while being computationally efficient, both in evaluation and sampling. These competing requirements define our design choices, which we motivate next.

7.4.2 Design of F

The design of F follows three principles: First, to achieve accurate renderings, we need it to be expressive enough to represent any complex probability density distribution. Second, we need an analytic form to compute its inverse F^{-1} . Finally, for efficiency, computing F , F^{-1} , and its Jacobian determinant needs to be as fast as possible.

Coupling Flows [DKB14; DSB16; KD18] achieve the efficiency requirements by using a single feed-forward pass of F and F^{-1} , and preserving a reduced level of computation, required to evaluate the Jacobian determinant, which is a lower triangular matrix for this type of flows. However, because all computation is done on a single pass to the models, coupling flows are typically less expressive than *Autoregressive Flows* [Mül+20; Kin+16; PPM17; Hua+18a; PSM19; Khe+21], which in turn, are less efficient during test time. We refer the reader to recent surveys [KPB20; Pap+21] for more comprehensive analyses of these topics.

To balance between efficiency and accuracy, we design F as a *Coupling Flow* and incorporate expressiveness through the coupling layer design. To this end, we evaluate several design choices available in the literature in the context of our problem. Recent work include *Affine Coupling* layers [DSB16], *Piecewise Linear* and *Piecewise Quadratic* [Mül+19], *Cubic-Spline Flows* [Dur+19b] and *Rational-Quadratic Spline Flows* [Dur+19a]. The type of coupling controls the

complexity of F , which can further be tuned by modifying the number of *coupling layers*, the *width* and *depth* of the neural network that define each coupling, and the number of *bins* which define the complexity of the polynomial function inside each coupling layer.

In addition, as in any neural network, it is possible to tweak the internal normalization layers, activation functions, optimizer type and configuration, regularization choices, etc. These create a combinatorial explosion of design choices, with two competing objectives of accuracy and efficiency. We found that *Rational-Quadratic Spline Flows* [Dur+19a] with a reduced amount of small coupling layers are expressive to a sufficient degree for every environment map that we tested, while being very efficient during evaluation, achieving orders of magnitude less computational cost than tabulated sampling. Because F is efficient and fully differentiable, it could potentially be incorporated into inverse rendering problems. We specify our model design choices in Section 7.6 and evaluate them in Section 7.8.1.

7.4.3 Training F

We train our normalizing flow to minimize the aggregated negative log-likelihood of a set of samples, $\mathcal{B} = \left\{ (\theta, \phi)^{(n)} \right\}_{n=1}^N$, that are drawn dynamically

in batches of size N from the input environment map I as described in Section 7.3.2.

During each training step, we minimize the average negative log likelihood across the elements in the batch $\frac{1}{N} \sum_b \mathcal{L}_{nll}(\theta, \phi)^{(b)}$ where:

$$\mathcal{L}_{nll}(\theta, \phi) = -\log(p_{\mathcal{F}}(\theta, \phi)) = -\log \left(p_z(\mathcal{F}^{-1}(\theta, \phi)) \left| \det \frac{\delta \mathcal{F}^{-1}}{\delta(\theta, \phi)} \right| \right) \quad (7.5)$$

This is done by learning to map from the distribution of samples to the noise distribution $p_z(z)$. Given a sufficient number of training batches, F learns to accurately sample from the distribution of samples encoded in the input environment map I . Because F is invertible and due to our efficient model design, the probability of a sample (θ, ϕ) can easily be computed by its inverse F^{-1} .

During training, we observe that most of the computation time is used to generate the ground truth batches of samples $\mathcal{B}$, which is the very process that we are interested in improving with F . We also notice that achieving a training procedure that works for every environment map is challenging due to gradient instabilities. We introduce a series of modifications to the training procedure to allow for stable training dynamics and avoid NaNs for any input environment map, which we specify in Section 7.6.

7.5 Environment Map Compression

In addition to learning to sample and evaluate the PDF, we train a compression neural network $\mathcal{C}$ which learns to map from directions to RGB values: $\mathcal{C}(\theta, \phi) = \text{RGB}$, $\theta \in [0, \pi)$, $\phi \in [0, 2\pi)$, $\text{RGB} \in \mathbb{R}^{+3}$. This serves two purposes: First, it provides a memory and time efficient representation of the environment map, thus further improving the render times and computational cost. Second, it transforms the tabulated representation into a continuous and differentiable function for which gradients can be computed, which may prove useful for inverse rendering scenarios. For the model design of $\mathcal{C}$, we build upon previous work on implicit neural representations. In particular, neural network architectures that work on the frequency domain have shown increased performance with respect to MLP for modeling natural signals [Tan+20; Sit+20], both in reconstruction quality and parameter efficiency. We thus model $\mathcal{C}$ as a shallow *sinusoidal MLP*, a *SIREN* [Sit+20]. This architectural design has been explored in recent work on environment map approximation, such as *RENI* [GES22]. However, as we do not aim to learn any prior over natural illumination, we can remove some constraints introduced in *RENI*, most notably the *rotation equivariance* requirement. Further, we do not use any latent vector to condition the output of our model, nor we do require *Vector Neurons* for our representation. With our simplifications, we achieve much higher reconstruction quality with the same parameter count. We train $\mathcal{C}$ on linear RGB space, with a pixel-wise reconstruction loss. Specifically, we use an L_1 loss, as it produces sharper and more accurate reconstructions than higher-order norms, such as L_2 [Iso+17; RG21]. We also observe strong gradient instabilities when training with L_2 on linear RGB. As $\mathcal{C}$ expects inputs in radians, but the input image has ranges of $H \times W$ pixels, our full loss becomes:

$$\mathcal{L}_{\text{recon}} = \frac{1}{H \times W} \sum_i^H \sum_j^W \left\| \mathcal{C} \left(\frac{i \times \pi}{H}, \frac{j \times 2\pi}{W} \right), \mathcal{I}(i, j) \right\|_1 \quad (7.6)$$

We use mini-batching to train our compression models $\mathcal{C}$, using uniformly sampled directions θ , ϕ . Previous work [GES22; Szt+21; Rod+23a; Lav+21] introduce adaptive sampling, cosine weighting or specular peak attenuation to similar loss functions for environment map or BRDF reconstruction. However, for our particular problem, we empirically observed that these additions tend to contribute negatively either to the training dynamics, or to the final quality of the reconstruction.

7.6 Implementation Details

Preprocessing Before the rotation algorithm described In Section 7.3.1, we resize every input HDRI environment map to a resolution of (2000, 4000) pixels using area interpolation.

This is an optional step, but it helps us standardize our experiments so that input resolution does not change training times. Note that evaluation times of our NEnv models depend on the model sizes and parameterizations, not on the resolution they were trained on.

Model Design and Training Our models sizes, loss functions, coupling layer design, normalization, optimizer and training configurations were selected using a combination of manual tuning and Bayesian hyperparameter optimization using *Weights and Biases* [Bie20] for a variety of representative environment maps.

We train the models using PyTorch [Pas+19] 1.11 and Torchvision [MR10]. Our flow and compression networks are trained independently, and individually for each input environment map. We empirically observe that mixed precision training [Mic+18] introduces strong training instabilities for this problem, so we train the models using *float32* precision. However, our models can be evaluated using half precision, which significantly increases their efficiency during test time. *F* and *C* each use around 3 MBs each, while a (2000, 4000) resolution HDRi environment map uses around 25 MBs.

Normalizing Flow Our normalizing flows *F* build upon the official implementation in [Dur+19a]. We use Batch Normalization [IS15] in the residual layers of our flows. To maximize evaluation efficiency, our flows have a limited number of trainable parameters. In particular, we use only 2 coupling layers, with 2 hidden layers, each with 256 hidden units and 256 *bins*. We use spline flows as our coupling layer type [Dur+19a]. Our initial learning rate is of $5e - 4$, which is halved every 2500 iterations. We use Adam [KB15] as our optimizer. To stabilize training, we use gradient norm clipping [Zha+20] ($maxnorm = 1$). This is crucial for well-behaved models and efficient convergence. We train our models with a batch size of 100000 for 15000 epochs, which takes around 2 hours on an Nvidia RTX 3060 GPU. Note that fewer iterations (eg 5000) are typically sufficient for adequate results, while longer training helps to resolve additional details.

Compression Network We use a small sinusoidal MLP as our environment map compression network *C*. We use the official implementation in [Sit+20]. We train our models with a batch size of 500000 of randomly sampled directions and linear RGB values, for 10000 epochs, which takes around 0.5 hours on an Nvidia RTX 3060 GPU. Our initial learning rate is $5e - 4$, which is halved every 2000 iterations. We use Adam [KB15] as our optimizer. To stabilize training, we use gradient norm clipping ($maxnorm = 50$). Our compression networks have 3 layers with 512 hidden units each. We initialize their weights as described in [Sit+20].

Rendering We use our own path tracer (PT) rendering pipeline as a baseline engine. This PT uses an implementation inspired by *Wavefront* [Van11;

LKA13], powered by *Embree* [Wal+14]. We distribute render work in tiles of (25,25) pixels that perform path-tracing in usual Wave-front steps (ray generation, intersection, shading, connection), each tile using a separate CPU thread. Note that tile size could be changed, as our selected tile dimensions are fixed due to hardware limitations and the amount of current thread active simultaneously (Intel Core i7-7700K CPU @ 4.20GHz, 8 threads).

Under this architecture, we perform our environment map sampling step once per tile, before every pixel in each tile performs its connection step. This way, we can either use a CPU approach [Cla+21] per pixel or integrate a single call to F to obtain all samples that are required in the current tile. We use *LibTorch* to use our PyTorch implementation from our C++ code. Intersections are performed in a single call using a ray packet call from Embree. Integrating C is more challenging, since we require ray data. However, we can store all this information simultaneously: We prepare all rays for intersection, and perform a single call to C . This way, we obtain all values at once. Note that these evaluations will be discarded if the next path-tracing bounce does not go out of bounds.

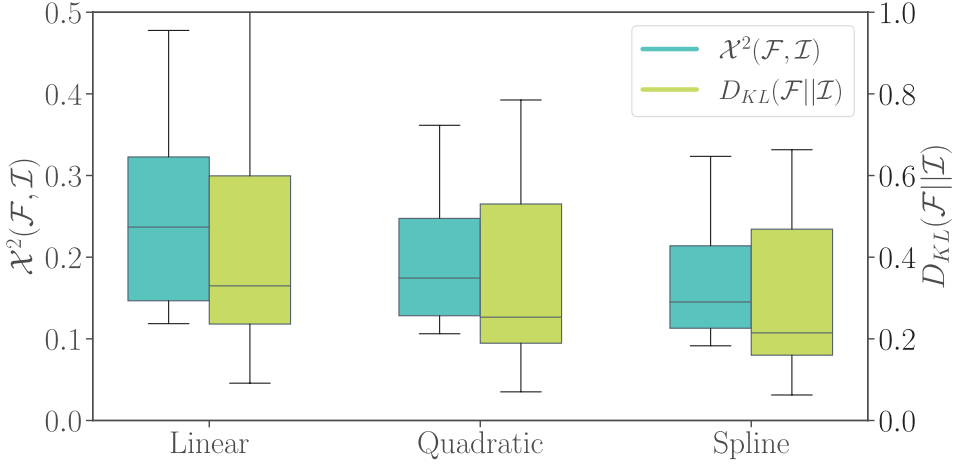
7.7 Dataset

Our goal is to propose a method that works for any type of global illumination which can be represented using environment maps. To this end, we gather a dataset of 32 high resolution HDRi environment maps from publicly available sources, notably [HDRMaps](#) and [Poly Haven](#), to thoroughly evaluate our models. Our dataset contains natural daylight, including sunny midday illumination, cloudy diffuse images, sunset, and dusk. Further, we also test our method on indoor scenes, from studio lighting, to a concert hall or cathedral illuminations. We purposefully include very challenging cases with multiple colored light sources, which helps us understand the limitations and capabilities of previous work and our models.

7.8 Evaluation

In this section, we evaluate our models, both quantitatively and qualitatively. We first measure their performance in terms of reconstructing the probability density and the RGB values of the input environment maps, and in computational cost. Finally, we measure their integrated quality in final rendered images compared to a variety of baselines. Unless stated otherwise, we use the full dataset described in Section 7.7 for our analyses.

Figure 7.5: PDF χ^2 and KL Divergence [KL51] of normalizing flows using different coupling layer types, including Linear and Quadratic Couplings from [Mül+19] and Spline Layers [Dur+19a].



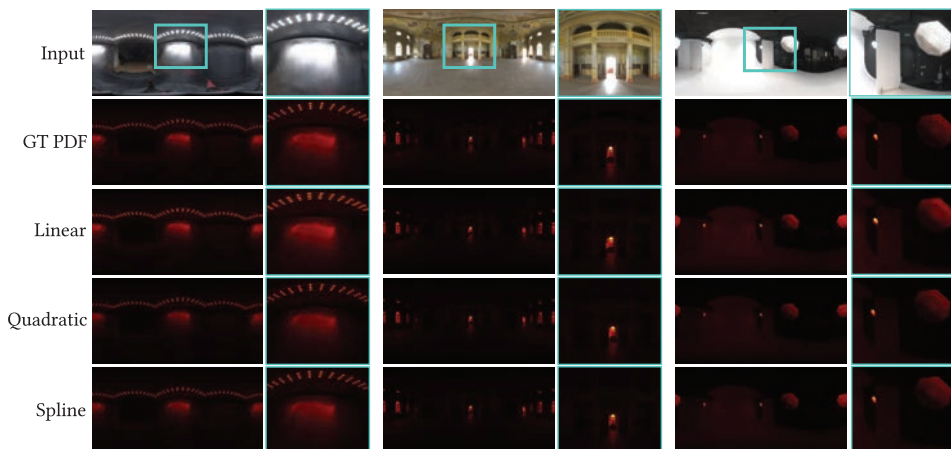
To make comparisons fair, we use the following configuration: For RENI [GES22], we train a new model from scratch with the exact same architecture design as our C , and train it for 3000 epochs. For Spherical Harmonics (SH) [See66; RH01b], we use 1024 coefficients, and for Spherical Gaussians (SG) [Xu+13], 16×32 dimensions. These configurations have a larger amount of parameters than what is typically used for these methods, especially in real time rendering. However, our goal is to maximize the reconstruction quality for each method and environment map, even if it results in an increased memory footprint. For the three methods we evaluate, the output resolution is set to (256, 512) pixels. We follow the official implementation of [GES22] for all these comparisons.

7.8.1 PDF Fit Accuracy

To validate our design choices for the coupling layers of our normalizing flow F , we train different versions of F for every environment map in our dataset, exclusively changing the type of polynomial function in the coupling layer. We measure the χ^2 distance, as well as the KL divergence $D_{KL}(F||I)$ [KL51] between the probabilities encoded by our normalizing flow p_F and the ground truth counterparts p_I as follows:

$$D_{KL}(\mathcal{F}||\mathcal{I}) = \sum_{\theta_i=0}^{\pi} \sum_{\phi_j=0}^{2\pi} p_{\mathcal{F}}(\theta_i, \phi_j) \log \left(\frac{p_{\mathcal{F}}(\theta_i, \phi_j)}{p_{\mathcal{I}}(\theta_i, \phi_j)} \right) \quad (7.7)$$

Figure 7.6: A qualitative comparison between the type of coupling layer used in our normalizing flows. On the top two rows, we show the input environment maps and their ground truth PDFs. Using the linear and quadratic couplings defined in [Mül+19] we achieve somewhat accurate encodings. With spline flows [Dur+19a], we achieve sharper and more accurate probability distributions. We use a $\gamma = 2.2$ to tonemap the RGB and PDF maps to help visualization. We highlight relevant regions using blue insets. Best viewed in color on a screen.

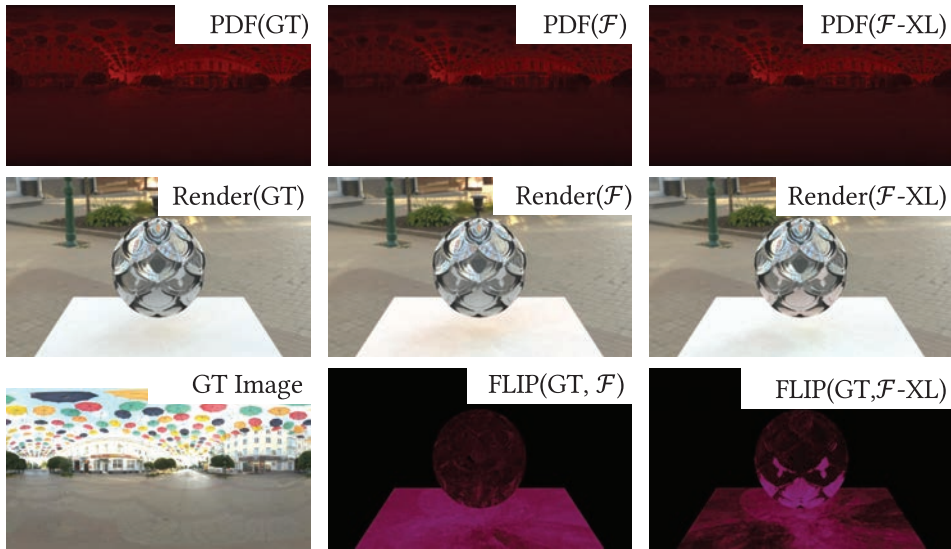


This metric, also used in [Mül+19], measures the distance between the learned probability distribution and the real PDF encoded in the environment map. In Figure 7.5, we show the aggregated divergences for our dataset for three different coupling layer types. As shown, the *Piecewise Linear* coupling proposed in [Mül+19] is outperformed by its more expressive *Piecewise Quadratic* variation, while the *Rational-Quadratic Spline Flows* proposed in [Dur+19a] achieve the lowest divergence overall, albeit by a relatively small margin. This improvement is achieved solely by adding expressiveness to the coupling layers, without any additional parameter cost. We provide a qualitative evaluation in Figure 7.6, where we show that linear and quadratic coupling layers encode less detailed probability distributions. These differences are particularly relevant for inputs with multiple distant light sources.

We also evaluate whether a larger model can yield significant improvements in accuracy. In Figure 7.7, we show the results of a large flow ($F\text{-XL}$), with twice as many layers and neurons per layer, a total of approximately 7 times more trainable parameters and model size in memory. Even though it indeed reduces the divergence from $D_{KL}(F||I) = 0.308$ to $D_{KL}(F\text{-XL}||I) = 0.281$, this model takes approximately one day to train and makes sampling 3.43 times slower. This larger model only

achieves a small gain in render accuracy (from 0.059 to 0.048 in FLIP [And+20]), suggesting that larger models exhibit diminishing returns in terms of quality. We believe our chosen model sizes adequately balance accuracy and speed. However, for practitioners that would like better accuracy at the cost of computational performance, we suggest using more coupling layers and more bins, as they tend to be the most effective design decisions beyond the coupling type.

Figure 7.7: A visualization of the impact of the size of F . $NEnv\text{-}XL$ is a larger version of F , with twice as many layers, bins and neurons per layer. This yields on small improvements on render and reconstruction quality, at the cost of sampling efficiency.



7.8.2 Environment Map Reconstruction

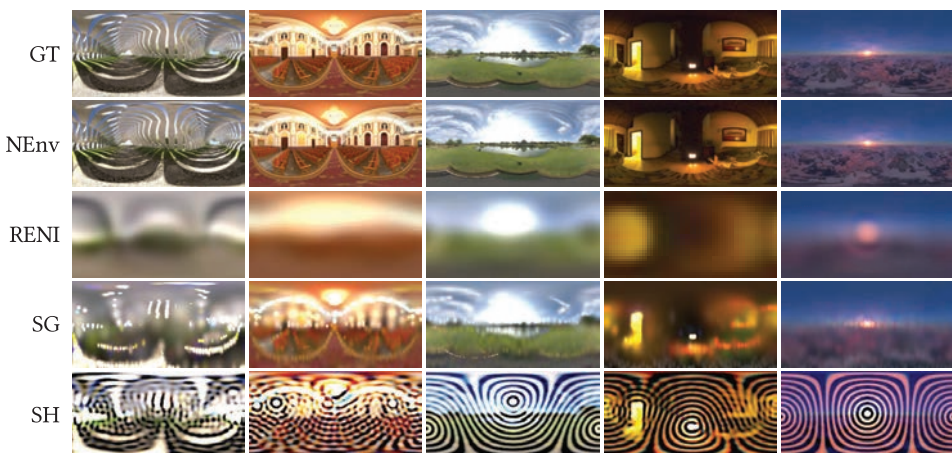
We now aim to measure the accuracy of the environment maps encoded by C and previous work on this problem. In Table 7.1, we show the average error on a variety of metrics for every environment map in our dataset. We use pixel-wise (PSNR) and perceptual metrics (SSIM [Wan+04], LPIPS-VGG [Zha+18], FLIP [And+20]) to thoroughly understand the differences between each method. We compute these metrics using PIQ [KZP19] and the implementation in [And+20], on images of (512, 256) pixels. We apply a $\gamma = 2.2$ to tonemap them before computing the distances. As shown, our network C clearly outperforms every other method, with SH struggling significantly, while RENI and SG achieve comparably better performances. We also provide a qualitative comparison in Figure 7.8, for a subset of our dataset, where it can be seen that C provides high-quality

reconstructions. RENI is optimized for outdoor, natural lighting, and it struggles with indoor illumination, providing overly smooth outputs. SG provides sharper results, while SH yields visually unpleasant results for every case: it is well known that the cyclic nature of SH tends to produce ringing artifacts, especially for high-contrast illumination environments and a large number of coefficients. We study the impact of these reconstructions on final renders in Section 7.8.4. Our representation c has the additional advantage that, thanks to its reconstruction quality, it can be used directly as background for samples that do not hit any geometry on the scene, while other alternatives require additional storage for higher quality versions, or even the original image.

Table 7.1: Average ($\pm$ std.) environment map reconstruction error for different methods, with pixel-wise and perceptual metrics. We use a color code to highlight **best** and **worst** cases.

Method	SH [RH01b]	SG [Xu+13]	RENI [GES22]	NEnv c (Ours)
PSNR $\uparrow$	13.68 $\pm$ 4.88	20.54 $\pm$ 5.04	18.92 $\pm$ 3.24	30.65 $\pm$ 5.59
SSIM [Wan+04] $\uparrow$	0.276 $\pm$ 0.20	0.559 $\pm$ 0.16	0.557 $\pm$ 0.16	0.873 $\pm$ 0.08
LPIPS [Zha+18] $\downarrow$	0.687 $\pm$ 0.07	0.637 $\pm$ 0.12	0.657 $\pm$ 0.12	0.155 $\pm$ 0.08
FLIP [And+20] $\downarrow$	0.505 $\pm$ 0.21	0.248 $\pm$ 0.11	0.278 $\pm$ 0.09	0.062 $\pm$ 0.03

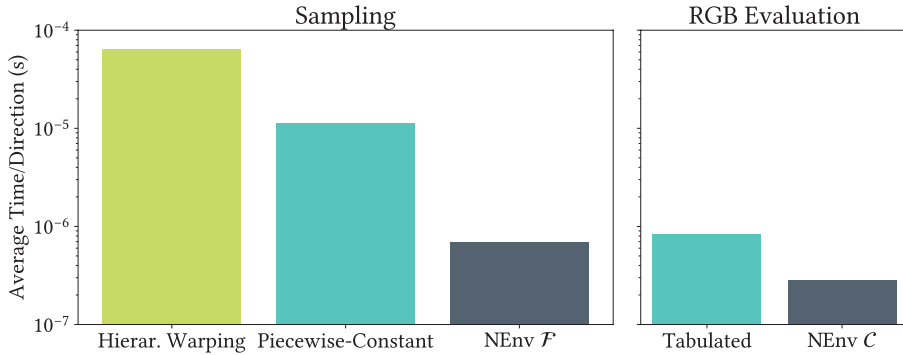
Figure 7.8: Environment maps encoded by our method and previous work, including RENI [GES22], Spherical Gaussian (SG) [Xu+13] and Spherical Harmonics (SH) [RH01b]. We use a $\gamma = 2.2$ to tonemap the RGB maps to help visualization.



7.8.3 Computational Cost

In Figure 7.9, we show the average time per sample obtained by the previous work and our normalizing flow F . On average, *Hierarchical Warping* [Cla+21; Pha19] obtains $6.48e^{-2} \pm 1.84e^{-3}$ milliseconds (ms) per sample, with the *PiecewiseConstant* method [PJH16] achieving $1.12e^{-2} \pm 3.24e^{-4}$ ms, while F obtains $6.83e^{-4} \pm 3.69e^{-6}$. To make these comparisons more favorable to each method, the timings for *Hierarchical Warping* [Cla+21; Pha19] and *PiecewiseConstant* [PJH20] are measured on CPU, since both algorithms rely on binary search which is difficult to parallelize on GPU, while the timings for NEnv are measured on GPU. For F , we use a sampling batch size of 200000, and we also include the times required to upload and download the samples from GPU, as in a real rendering scenario. We measure these times on the same hardware to make results comparable.

Figure 7.9: On the left, average seconds per sample for the baseline sampling algorithms and our model F . On the right, evaluation times for a tabulated baseline and our C . Note that we use a logarithmic scale for this plot.



While being differentiable and losing little reconstruction accuracy, F achieves, on average, 16.83 times more samples per second than the *Piecewise-Constant* method and 94.85 than *Hierarchical Warping*, with a maximum difference of 106.06. Furthermore, the timings for the *Piecewise-Constant* and *Hierarchical Warping* methods vary up to 15% and 12.5% respectively depending on the content of the environment map, while the variance of F is negligible. Not only did we achieve up to two orders of magnitude faster sampling than previous work, our method was also significantly more consistent in terms of timings. We found that *Hierarchical Warping* and the *Piecewise-Constant* methods typically struggle more with environment maps with multiple, distant light sources, while they are more efficient for natural daylight illumination.

In terms of RGB evaluation, a tabulated version of the environment map requires an average of $8.43\text{e}^{-4} \pm 1.44\text{e}^{-5}$ ms per direction, while our model $2.83\text{e}^{-4} \pm 7.85\text{e}^{-6}$ ms. Note that we use *TorchScript* with *Pytorch 1.11* to measure the times for C and F . Newer versions of these libraries, which allow for compiled models, will likely help achieve faster evaluations without changing the model architectures.

7.8.4 Rendering Comparisons

The goal of NEnv is to generate accurate yet fast environment map representations that do not make compromises in terms of rendering quality. We use the *Piecewise-Constant* PDFs as our ground truth (GT) and perform qualitative and quantitative analyses to show the reconstruction quality of NEnv with respect to previous work. We provide a fine grained analysis of NEnv, in which we evaluate each of the models separately and a full version of the model which uses both F and C . We use our path tracing engine configured as explained in Section 7.6 to render the same scene for each environment map. Each scene is rendered using a simple pinhole camera (no Depth of Field) and the environment maps as the only source of light, so we can reduce as much as possible external sources of Monte Carlo noise at low sampling. We use MIS, so both sampling and PDF functions for our maps impact the resulting image. All images are rendered with a resolution of (1080, 1920) pixels, 32 samples, maximum depth of 16, and path throughput weight Russian Roulette [PJH16].

In Figure 7.10, we show a qualitative comparison between different methods on the same scene, illuminated with different environment maps. We use an object from [Sch+17], which helps in highlighting the differences between far field illumination representations. As shown, our method, in any of their configurations, generates images that closely match the ground truth, while previous work struggles significantly, particularly in indoor illumination. Finally, in Table 7.2, we show a quantitative comparison of the average error obtained by these methods, for our entire dataset. For this analysis, we only measure the error on the object and floor on the renders in Figure 7.10, masking out the background. We observe that, even when combining our sampling and compression networks, our reconstruction quality is much higher than any of the methods in the previous work. Interestingly, the differences in environment map reconstruction quality we measured in Table 7.1, which were very unfavorable to SH, are not exactly correlated to rendering error, suggesting that SH, while being a pixel-wise inaccurate image representation, it tends to be precise on the more relevant information of the image, *i.e.*, where the key light sources lie. In terms of rendering, neither of the previous methods that we tested behaved consistently better than any other across every metric.

7.9 Conclusions

In this work, we have presented a neural rendering method for joint compression, evaluation, and sampling of environment maps for global illumination. We have validated our models quantitatively using a diverse dataset of environment maps. Our proposed lightweight normalizing flows provide accurate yet efficient sampling and pdf evaluation, achieving significantly faster computational times than analytical approaches with negligible loss in quality. Further, our sinusoidal compression networks achieve high-quality environment map approximations, surpassing previous work in both quality and generality. Every component in our models is differentiable, and they can be seamlessly integrated into engines like Mitsuba [Nim+19; Jak+22] or PRDPT [FR22] for inverse rendering applications. All in all, NEnv can represent global illumination with higher generality, quality and computational efficiency than previous work, with fully differentiable components.

Table 7.2: Average ($\pm$ std.) render reconstruction error for different methods, with pixel-wise and perceptual metrics. We use a color code to highlight **best** and **worst** cases.

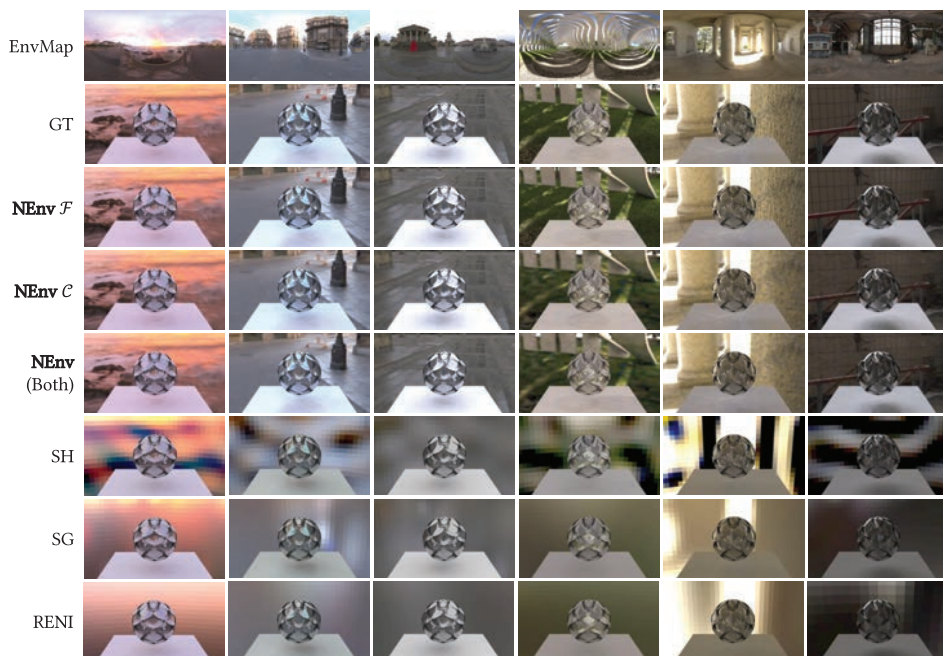
	PSNR $\uparrow$	SSIM [Wan+04] $\uparrow$	LPIPS [Zha+18] $\downarrow$	FLIP [And+20] $\downarrow$
SH [RH01b]	21.77 $\pm$ 5.35	0.933 $\pm$ 0.05	0.160 $\pm$ 0.03	0.150 $\pm$ 0.06
SG [Xu+13]	21.62 $\pm$ 6.34	0.940 $\pm$ 0.05	0.153 $\pm$ 0.03	0.154 $\pm$ 0.07
RENI [GES22]	21.53 $\pm$ 5.62	0.936 $\pm$ 0.04	0.159 $\pm$ 0.03	0.151 $\pm$ 0.05
NEnv (F)	37.97 $\pm$ 3.73	0.996 $\pm$ 0.00	0.024 $\pm$ 0.01	0.032 $\pm$ 0.02
NEnv (c)	41.05 $\pm$ 8.89	0.997 $\pm$ 0.01	0.016 $\pm$ 0.01	0.036 $\pm$ 0.02
NEnv (Both)	35.16 $\pm$ 4.97	0.994 $\pm$ 0.01	0.033 $\pm$ 0.01	0.049 $\pm$ 0.02

Our work could be extended in several ways. We require two networks, one for sampling and pdf evaluation, and another for compression. While this dual approach allows us to maximize their individual efficiency, a future research avenue is to build a single model which can solve the three tasks at the same time, to simplify our representation and possibly help with parameter reuse across models. However, fusing a sinusoidal MLP with a spline-based normalizing flow is not straightforward and requires a significant research effort that lies outside of the scope of this work. Recent work on architectural design [Meh+21] may provide cues on how this could be achieved. Further, we require to train new models for each input environment map. Building upon

recent work [GES22; Rai+22; GMX22; Szt+21; Hu+20], we could build a prior over environment maps, which should help reduce training times and solve inverse illumination problems, as in [Li+20; Azi+19; HHM22; YS19; Wan+21; YS21; Sri+20; Gar+19; Wan+23]. Achieving this without losing computational efficiency is a challenging future research problem. Besides, recent work on neural rendering [Mül+22; Che+22; Wan+22d; Att+22; Liu+22a] has shown that with carefully designed CUDA kernels or architectural representations, it is possible to achieve more efficient training and evaluation times. Integrating these ideas into our framework could help further increase its computational efficiency.

Finally, we believe that the ideas we present in this work can be used for other path tracing and physically based rendering problems, like aggregate scattering [Blu+16] or BxDF representations [Szt+21; CNN22; Zha+21; Che+20a]. We hope our work inspires future work on generative models for physically-based and neural rendering.

Figure 7.10: Render comparisons of different configurations of our work with the ground truth environment map and previous work. Best viewed in color on a screen. Please zoom in for details.



CHAPTER 8. CONCLUSIONS

Throughout this thesis, we have addressed important challenges in the fields of digital materials and radiance encoding, including *Visual Attribute Transfer* (Chapter 2), *SVBSDF Capture* (Chapter 2.C), *Tileable Texture Synthesis* (Chapter 3), *Neural BTF Propagation and Encoding* (Chapter 4), *SVBRDF Estimation* (Chapter 5), *Fabric Mechanical Parameter Capture* (Chapter 6) and *Neural Methods for Global Illumination* (Chapter 7). In this final section, our goal is to provide a comprehensive summary of the contributions made in this thesis and their significance in the fields of computer vision, graphics, and machine learning. By providing a broader view, we aim to highlight the importance and impact of the work presented here, both in terms of advancing the state-of-the-art and in its potential for real-world applications.

Digital Material Generation

The first projects developed in the context of this thesis are concerned with generating high quality digital materials, which can then be used for other tasks, like training neural material representations or creating large datasets for material estimation models. These methods take as input a single material, which helps increase the predictability of their outputs.

In Chapter 2, we present a neural visual attribute transfer framework capable of transferring, for a given material, many types of visual property maps to images of new patches of the same (or similar) material. These novel images may be taken under any illumination conditions, color variations, affine distortions, or camera setup. This model uses lightweight image-to-image convolutional neural networks to provide the first solution capable of leveraging the optical behavior of the material for the visual attribute transfer problem, by being trained on with a photometric dataset and a comprehensive data augmentation policy. This method can be trained in less than one minute, and is capable of transferring any type of visual attribute, from surface normals to stylizations or semantic segmentation maps. However, it is limited in accuracy, can only learn from one dataset at once, and can only learn to transfer one visual attribute.

To address these shortcomings, we extend this method in Chapter 2.C, where we propose a neural network capable of generating high-resolution large-scale SVBSDF maps of any material given one or more photometric datasets which represent small portions of a material. By extensive changes to the training procedure and improvements to the model architecture, not only do we enable this visual attribute transfer with the capability to learn from multiple datasets and transfer multiple images simultaneously, but we also increase the accuracy

and sharpness of its estimations. However, these improvements increase the computational cost of training the neural network, and limit their generality. With these models, we can build large-scale materials, which can be used to generate BTF data to train other representations (Chapter 4) or build large datasets to train models that generalize to new materials (Chapter 5).

Figure 8.1: Renders generated with SEDDI's [Textura.ai](#) materials, estimated with methods derived from some of the contents in this thesis.



Textures used in rendering applications also need to be tileable. That is, when they are spatially concatenated, the seams between tiles are not visible. In Chapter 3, we introduce a single-image Generative Adversarial Network (GAN) which can generate multiple tileable textures of the same material. Our method allows for tileable outputs using a novel latent space manipulation algorithm, and handles multiple texture maps simultaneously, through neural architecture improvements. Further, we leverage the GAN's discriminator as a test-time quality estimation function, which helps distinguish between high quality and low quality outputs. This model achieves higher quality outputs than previous work, at lower computational times.

These models rely on a single material as input, which, while increasing predictability, limits their generalization capabilities. Nonetheless, our findings provide evidence that under certain assumptions and highly controlled training procedures, neural networks can be utilized for generating ground truth data. This data can be leveraged for other tasks, as demonstrated in other sections of this thesis.

Neural Radiance Encoding

One of the research problems that we explore in this thesis is the challenge of representing radiance using neural networks, which offers several potential

benefits compared to traditional approaches. Neural networks can provide efficient, continuous, and fully differentiable functions that are useful for forward and inverse rendering. Additionally, they enable novel editing capabilities and efficient sampling for Monte Carlo rendering. In this work, we focus on two key components of virtual scenes: materials and illumination. Our goal is to develop neural network-based approaches for encoding these elements that are both accurate and computationally efficient, paving the way for new applications in computer graphics and vision.

For materials, in Chapter 4, we introduce a novel neural representation for encoding reflectance. Building upon previous work on the topic of neural materials [Kuz+21], our method has three components: A *neural texture*, which encodes the reflectance of the material in a 2D latent space; a fully-convolutional *neural renderer*, which takes as input this latent reflectance and light and view angles and outputs linear RGB values; and an *autoencoder*, which can estimate the neural texture of any input 2D image of the material. Similarly to previous work on neural fields [Mil+20], we train each model from scratch for every material, taking as input its *Bidirectional Texture Function* (BTF), which can be either captured or synthetically generated. Our approach introduces the first neural BTF representation with conditional input, which makes it possible to extrapolate BTF measurements, create tileable BTFs, and synthesize new materials through reflectance propagation. We address one of the main limitations of prior work on neural materials, which was their lack of editing capabilities, hindering their applicability to real-world rendering scenarios that require tileable and large-scale materials. Despite these improvements, our representation has limitations in that semantic properties such as specularities or albedo are encoded on the latent space and thus not easily editable. However, we believe that our method has significant potential for future research on neural material representations.

In Chapter 7, we addressed the problem of illumination encoding, introducing a novel neural method for joint sampling, PDF encoding, and RGB compression of environment maps used for global illumination in rendering. Our method builds on previous work on *implicit neural representations* for compressing the HDRI environment map into a continuous, differentiable function. We designed a *Normalizing Flow* that can sample directions from environment maps and measure their probability density, both of which are essential for Multiple Importance Sampling in Monte Carlo rendering. Our carefully designed neural architecture and training procedures allowed us to obtain models that are up to two orders of magnitude faster than analytical methods while still being fully differentiable. Using a dataset of environment maps with diverse properties, we demonstrated high generality and accuracy, outperforming previous work on the topic. To encourage future research, we will provide an open-source implementation and a dataset of trained models. While this is the first method that learns to sample from environment maps, we believe there is potential

to further improve our approach by introducing more sophisticated neural network designs or learning priors over global illumination.

Despite their shortcomings, these two neural methods introduce contributions to the field of neural scene representations, and they could both be incorporated into neural rendering pipelines and more traditional path-tracing engines. We hope these ideas inspire further research interest in this field.

Scalable Material Digitization

One of the key objectives of this thesis is to develop digitization solutions that are both scalable and affordable, without relying on expensive or inaccessible hardware. To accomplish this, we have made two different contributions. First, in Chapter 5, we introduce a novel method for estimating *spatially varying bidirectional reflectance distribution functions* (SVBRDFs) using flatbed scanners as the capture device. This method strongly relies on the data generation algorithms presented in previous sections of this thesis. Our approach employs a custom-built, lightweight neural network with attention mechanisms, resulting in estimations with considerably higher resolution than previous methods that relied on smartphone flash-lit images as input. We empirically observe that material reflectance is strongly dependent on its microgeometry, which flatbed scanners can capture appropriately, all the while providing images that can be directly used as the albedo of the SVBRDF. One of the key contributions of this work is the introduction of the first uncertainty quantification algorithm for material estimation, for which we demonstrate applications on dataset creation through active learning. Further, an extension of this work is part of [Textura.ai](https://textura.ai), which is utilized by real-world users, providing evidence of its robustness and reliability. In Figure 8.1, we show some digital garments which use materials digitized with this technology. We are confident that the novelties introduced by this method will shape future research on the long-standing problem of single image material digitization. Although we are confident in the method's capabilities, it can be extended to estimate additional reflectance properties like anisotropy or transmittance, to generalize to other materials or devices, and to improve its accuracy even further.

Besides, in Chapter 6, we proposed a method for mechanical parameter estimation of textile materials using depth images and the material density as inputs. Our method achieves greater scalability than previous work, which required video sequences, manual intervention, or purposefully built and expensive devices. To do so, we propose a solution that is agnostic to the optical appearance of the fabric, and can be utilized by non-expert operators. Our custom-built multi-input neural network predicts the full set of mechanical parameters of any textile using synthetic data, leveraging a strong data

augmentation policy, pre-training, attention mechanisms, and multiple forms of pooling of internal activations. A significant contribution of this work is our proposed *drape similarity metric*, which measures distances between the mechanical behavior of fabrics while using only images as input. We validate this metric with a user study conducted on-site, showing that it strongly correlates with human perception. Although our method has significant contributions to estimating the mechanical behavior of textiles, it is still limited in at least two ways: it takes depth images as input, which are relatively easy to obtain but are less accessible than RGB pictures, and the proposed drape similarity metric is not easily differentiable, hindering its potential as a loss function in optimization problems.

These two estimation algorithms, along with the uncertainty quantification algorithm and the proposed drape similarity metric, have the potential to transform the low cost material digitization field, introducing more scalable, robust, and perceptually-validated solutions.

New Frontiers

Although this thesis has made significant contributions to the fields of radiance encoding, and digital material capture and edition, we acknowledge that our proposed methods have certain limitations in terms of scope and generality. Additionally, concurrent developments in deep learning, neural rendering, and computer vision have opened up new and exciting research avenues that will shape the future of the field. With this in mind, we would like to highlight several research directions that we believe are particularly intriguing for extending the work presented in this thesis.

Generative Models for Materials

In this thesis, we have presented models that are limited to solving individual tasks such as material digitization, encoding, or tileable texture synthesis. However, recent research [Rom+22; HJA20; Sah+22] has demonstrated that a single generative model can be used to solve multiple tasks simultaneously. Developing a single generative model for materials that can be used for capture, super-resolution, edition, real-time tileable synthesis, or material interpolation would be an exciting research direction for the future of digital materials. One of the main challenges in this area is the lack of publicly available datasets of real materials, which are essential for reproducible research on deep learning algorithms. Nevertheless, techniques like pre-training, fine-tuning, and data augmentation policies may be employed to construct these models without requiring massive datasets of materials.

Multimodal Learning

The findings in this thesis suggest an interesting research avenue: multimodal learning for materials. In addition to capturing a material's reflectance and mechanical properties, it is possible to digitize other attributes such as its fabrication process, semantic attributes, fine-grained composition, origin, cost, or even its carbon footprint. Recent work on multimodal and contrastive learning [Par+20] suggests that a shared embedding for visual and semantic attributes can be found, which can then be utilized for various tasks such as zero-shot learning or conditioning generative models [Rom+22]. Developing such an embedding for materials could be extremely valuable for a wide range of problems, including material edition, interpolation, classification, generation, similarity measurement, or anomaly detection. This research direction is promising, but it also poses significant challenges, such as the need for large and diverse datasets that include multiple modalities of material information, and the development of effective multimodal learning architectures that can handle such information. Nonetheless, we believe that the benefits of multimodal learning for materials make it a hugely compelling direction for future research.

Neural Spatially-Varying BSDFs

In Chapter 7, we demonstrated the effectiveness of invertible generative neural networks for efficient Multiple Importance Sampling in global illumination. Similar networks have also been utilized for BRDFs [Szt+21], suggesting that learned sampling could be applied to spatially-varying materials. Further, in Chapter 4, we propose a neural representation of spatially-varying materials, which showed promising compression results. Combining the ideas of these two projects, it should be possible to train invertible neural networks to learn to sample from real-world materials, which often exhibit complex optical behavior such as spatially-varying reflectance or transmittance, we can potentially improve the efficiency and quality of path-traced renders while enabling new inverse rendering applications. This presents a promising direction for future research in material capture and rendering.

Uncertainty Quantification and Model Understanding

We have presented the first method for uncertainty quantification in material capture applications, demonstrating its utility for active learning and dataset creation. Additionally, we have shown that visualization of material capture models can aid in understanding their predictions. However, we believe that these areas are still relatively unexplored, and further analysis could provide valuable insights for designing the next generation of material digitization

pipelines. This includes developing more suitable capture devices, identifying valuable data points for labeling, and building more accurate models. Furthermore, we believe that investigating methods for communicating to end users when a capture may be inaccurate and suggesting alternative digitization pathways is an important research topic that could enhance the impact, robustness, and trustworthiness of these models in the industry.

Enhancing Digitization Quality and Realism

In this thesis, we have presented methods for highly realistic material encodings (Chapter 4) and digitization pipelines (Chapter 2.C). They can be used to represent a wide variety of materials at very high qualities, however, they require expensive capture systems. On the other hand, in Chapter 5, we introduce a single-image digitization system capable of capturing a small set of SVBRDF parameters. While the model is adequately accurate, this limited material model cannot represent complex optical phenomena, like anisotropy or transmittance. Extending single-image material digitization systems to be able to accurately estimate more reflectance properties is a very challenging research problem, because this estimation is inherently ill-posed, and there is very limited available training data. These low cost capture systems will be able to increase the realism of their digitizations in the future, as more and better data becomes available, and better models can be built as the machine learning field advances.

Enhancing Generality on Devices and Materials

A limitation of our casual capture algorithms is that they require the use of specific, albeit low-cost, devices such as flatbed scanners or depth cameras. An immediate extension of our methods is to make them compatible with other types of devices, particularly smartphone cameras, which would increase their applicability. However, less controlled capture settings typically hinder the accuracy of digitizations, as additional assumptions must be made to obtain a plausible result. With the ongoing improvement in the quality of smartphone cameras, along with larger datasets and better deep learning models, high-quality digitization of material reflectance or fabric mechanics using smartphones could become a reality in the near future.

In addition, like any learning-based system, our models are constrained by the materials present in their training set, limiting their ability to generalize to other materials. Although we have achieved accurate digitization of fabrics and leathers, there is no guarantee that our models will perform well on out-of-distribution materials such as stone, plastics, metals, or glass. While some of these materials may be successfully digitized using single-image

models with sufficient data, others may present more significant challenges. It is our hope that future research in this area will help improve the generality and accuracy of these models.

Digitization on Edge Devices

One limitation of material capture algorithms based on deep learning is that they require expensive hardware for inference. This means that in commercial products, users can capture images on a commodity device, but the digitization process is typically carried out on a cloud-based application. As a result, bottlenecks may arise that hinder the efficiency of the digitization process, with users having to wait several minutes before obtaining their digital material. To achieve real-time learning-based digitization, it may be necessary to perform model inference on edge devices such as smartphones, tablets, or laptops. However, this approach may place limitations on model sizes, potentially leading to a reduction in digitization accuracy. Recent work on edge deep learning [Wan+20c] may provide interesting cues for future research on this topic.

Neural Scene Representations

Concurrently to the work in this thesis, *Neural Radiance Fields* [Mil+20] emerged as an alternative powerful representation for virtual scenes. A Cambrian explosion of research in radiance fields for scene representations has emerged to increase their capabilities, in terms of data efficiency, speed, realism, or generality [Wan+22d; Mül+22; Wan+23; Che+22; Tew+22; Ver+22; Fri+22; Att+22; Xie+22]. However, these remain somewhat limited in controllable and capabilities for edition, for which more traditional rendering and scene creation pipelines may be more adequate than learning-based alternatives. Our neural encodings for illumination or materials, which may be integrated into either neural representations or path-traced rendering engines, may help bridge these two pipelines, which should benefit both.

Dataset Generation for Vision Downstream Tasks

Synthetic data has been extensively shown to help deep learning models generalize to real images [Azi+23; Zhu+21; Ric+16]. Our proposed method for casual capture of mechanical parameters in fabrics (Chapter 6, [Rod+23b]) provides additional evidence in this direction. The procedural creation of virtual environments for pre-training of computer vision models will remain a very relevant topic in future years in the field, as it will reduce the need for expensive labeling of real-world data. These virtual environments strongly

benefit from highly accurate assets for enhanced realism, as well as efficient yet expressive image formation algorithms. We hope that the contributions in this thesis will help in generating synthetic data for training computer vision models which solve general tasks, including classification or segmentation [Kir+23].

Applications to Other Fields

Finally, a fascinating use of computer vision and material digitization is *digital cultural heritage* [HMY09]. Digital capture of artifacts from ancient civilizations could help important research fields, like anthropology or archaeology. However, past solutions required inaccessible devices [Ham+21; Dye+18], which limited their impact. Inexpensive gathering of photorealistic copies of ancient artifacts, which can be shared digitally, can provide valuable solutions for these research fields. We hope that low cost and accurate digitization systems may not only help the visual media and design industries, but also other fields for which digital asset inventories are helpful.

Closing Remarks

The collaborative efforts between disciplines and research institutions, both public and industrial, have been instrumental in enabling the contributions of this thesis, and the advancements made in the computer vision, graphics, and machine learning fields as a whole in the past few years. The interdisciplinary environment in which I worked during my PhD greatly enhanced the quality of my research and communication skills, while also providing me with the opportunity to learn fascinating new topics and collaborate with exceptional people. Further, while the massive amounts of publications in these fields put a significant amount of pressure on the researchers, it makes the field move incredibly quickly and creates a very exciting environment to work in. I hope that the focus on open science and the integration of ideas and feedback from other research areas continues, along with a greater awareness of the ethical and ecological impact of the models we create and deploy. Working as a researcher is in many ways a privilege, and it is our responsibility to advance science in a way that positively impacts society and the environment, and creates a better, more prosperous, sustainable, and equitable future for everyone. Looking ahead to the future, my hope is that the contributions made in this thesis will have a tangible impact on the real world and serve to further advance research in positive ways. On a personal level, my goal is to continue learning and challenging myself, all while striving to make a positive difference in the world.

BIBLIOGRAPHY

- [Aba+16] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. "Tensorflow: A System for Large-Scale Machine Learning". In: *12th Symposium on Operating Systems Design and Implementation*. 2016, pp. 265–283 (cit. on p. 96).
- [Aga+03] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. "Structured Importance Sampling of Environment Maps". In: *ACM Transactions on Graphics (TOG)* 22.3 (2003), pp. 605–612 (cit. on p. 185).
- [AAL16] Miika Aittala, Timo Aila, and Jaakko Lehtinen. "Reflectance Modeling by Neural Texture Synthesis". In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), pp. 1–13 (cit. on p. 124).
- [AWL13] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. "Practical SVBRDF Capture in the Frequency Domain". In: *ACM Transactions on Graphics (TOG)* 32.4 (2013), pp. 110–1 (cit. on pp. 17, 107, 128, 258).
- [AWL15] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. "Two-shot SVBRDF Capture for Stationary Materials". In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 1–13 (cit. on pp. 17, 34, 104, 125, 258).
- [AKI+18] Adib Akl, Charles Yaacoub, Marc Donias, Jean-Pierre Da Costa, and Christian Germain. "A Survey of Exemplar-Based Texture Synthesis Methods". In: *Computer Vision and Image Understanding* 172 (2018), pp. 12–24 (cit. on p. 73).
- [Alc+19] Raul Alcaín, Carlos Heras, Iñigo Salinas, Jorge Lopez-Moreno, and Carlos Aliaga. "Microscale Optical Capture System for Digital Fabric Recreation". In: *Proceedings of the 7th International Conference on Photonics, Optics and Laser Technology - Volume 1: PHOTOPTICS*, INSTICC. SciTePress, 2019, pp. 114–119 (cit. on pp. 35, 39).
- [Alm+18] Amjad Almahairi, Sai Rajeshwar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. "Augmented Cyclegan: Learning Many-to-Many Mappings from Unpaired Data". In: *International Conference on Machine Learning (ICML)*. PMLR. 2018, pp. 195–204 (cit. on p. 85).
- [Ami+20] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. "Deep Evidential Regression". In: *Advances in*

- Neural Information Processing Systems* 33 (2020), pp. 14927–14937 (cit. on p. 126).
- [AP08] Xiaobo An and Fabio Pellacini. "AppProp: All-pairs Appearance-Space Edit Propagation". In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), pp. 1–9 (cit. on p. 33).
- [And+20] Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D Fairchild. "FLIP: A Difference Evaluator for Alternating Images." In: *Proc. ACM Comput. Graph. Interact. Tech.* 3.2 (2020), pp. 15–1 (cit. on pp. 198, 199, 202).
- [ASE17] Antreas Antoniou, Amos Storkey, and Harrison Edwards. "Data Augmentation Generative Adversarial Networks". In: *arXiv preprint arXiv:1711.04340* (2017) (cit. on p. 45).
- [ACB19] Devansh Arpit, Victor Campos, and Yoshua Bengio. "How to Initialize Your Network? Robust Initialization for Weightnorm & Resnets". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on pp. 109, 119).
- [ARV19] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. "A Critical Analysis of Self-Supervision, or What We Can Learn from a Single Image". In: *International Conference on Learning Representations (ICLR)*. 2019 (cit. on p. 75).
- [Att+22] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. "Learning Neural Light Fields with Ray-space Embedding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 19819–19829 (cit. on pp. 187, 203, 212).
- [Azi+19] Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. "Inverse Path Tracing for Joint Material and Lighting Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2447–2456 (cit. on p. 203).
- [Azi+23] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. "Synthetic Data from Diffusion Models Improves ImageNet Classification". In: *arXiv preprint arXiv:2304.08466* (2023) (cit. on p. 212).
- [BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer Normalization". In: *arXiv preprint arXiv:1607.06450* (2016) (cit. on pp. 109, 111, 119, 176).
- [Baa+22] Hendrik Baatz, Jonathan Granskog, Marios Papas, Fabrice Rousselle, and Jan Novák. "NeRF-TeX: Neural Reflectance

- Field Textures". In: *Computer Graphics Forum*. Vol. 41. 6. Wiley Online Library. 2022, pp. 287–301 (cit. on pp. 104, 187).
- [BSK23] Steve Bako, Pradeep Sen, and Anton Kaplanyan. "Deep Appearance Prefiltering". In: *ACM Transactions on Graphics (TOG)* 42.2 (2023), pp. 1–23 (cit. on p. 105).
- [BYK21] Wentao Bao, Qi Yu, and Yu Kong. "Evidential Deep Learning for Open Set Action Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 13349–13358 (cit. on p. 130).
- [Bar+09] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing". In: *ACM Transactions on Graphics (TOG)* 28.3 (2009), p. 24 (cit. on p. 74).
- [Bar+15] Connelly Barnes, Fang-Lue Zhang, Liming Lou, Xian Wu, and Shi-Min Hu. "Patchtable: Efficient Patch Queries for Large Datasets and Applications". In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 1–10 (cit. on pp. 32, 74).
- [Bar+21] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. "Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021 (cit. on p. 186).
- [Ben+21] Saguy Benaïm, Ron Mokady, Amit Bermano, and Lior Wolf. "Structural Analogy from a Single Image Pair". In: *Computer Graphics Forum*. Vol. 40. 1. Wiley Online Library. 2021, pp. 249–265 (cit. on pp. 31, 33, 46, 47, 56, 57, 60, 75).
- [Bén+13] Pierre Bénéard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare, and Katherine Breeden. "Stylizing Animation by Example". In: *ACM Transactions on Graphics (TOG)* 32.4 (2013), pp. 1–12 (cit. on p. 32).
- [Ben+20] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. "Learning Invariances in Neural Networks". In: *arXiv preprint arXiv:2010.11882* (2020) (cit. on p. 43).
- [BJV17] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. "Learning Texture Manifolds with the Periodic Spatial GAN". In: *International Conference on Machine Learning (ICML)*. 2017, pp. 469–477 (cit. on pp. 72, 74–76, 89–91, 95, 99).
- [Ber+23] Hugo Bertiche, Niloy J Mitra, Kuldeep Kulkarni, Chun-Hao Paul Huang, Tuanfeng Y Wang, Meysam Madadi, Sergio

- Escalera, and Duygu Ceylan. "Blowing in the Wind: CycleNet for Human Cinemagraphs from Still Images". In: *arXiv preprint arXiv:2303.08639* (2023) (cit. on pp. 17, 259).
- [Bha+03] Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popovic, and Steven M. Seitz. "Estimating Cloth Simulation Parameters from Video". In: *Symposium on Computer Animation*. The Eurographics Association, 2003 (cit. on pp. 156, 158).
- [Bi+18] Wenyan Bi, Peiran Jin, Hendrikje Nienborg, and Bei Xiao. "Estimating Mechanical Properties of Cloth from Videos Using Dense Motion Trajectories: Human Psychophysics and Machine Learning". In: *Journal of vision* 18.5 (2018), pp. 12–12 (cit. on p. 159).
- [Bie20] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020 (cit. on pp. 108, 176, 194).
- [Bit+20] Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. "Spatiotemporal Reservoir Resampling for Real-time Ray Tracing with Dynamic Direct Lighting". In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39.4 (2020) (cit. on p. 184).
- [Blu+16] Adrian Blumer, Jan Novák, Ralf Habel, Derek Nowrouzezahrai, and Wojciech Jarosz. "Reduced Aggregate Scattering Operators for Path Tracing". In: *Computer Graphics Forum (Proceedings of Pacific Graphics)* 35.7 (2016), pp. 461–473 (cit. on p. 203).
- [Bom+21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. "On the Opportunities and Risks of Foundation Models". In: (2021) (cit. on p. 159).
- [Bos+20] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik Lensch, and Jan Kautz. "Two-shot Spatially-Varying BRDF and Shape Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 3982–3991 (cit. on p. 125).
- [Bou+13] Katherine L Bouman, Bei Xiao, Peter Battaglia, and William T Freeman. "Estimating the Material Properties of Fabric from Video". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 1984–1991 (cit. on pp. 156, 159).

- [BS12] Brent Burley and Walt Disney Animation Studios. "Physically-based Shading at Disney". In: *ACM SIGGRAPH*. Vol. 2012. vol. 2012. 2012, pp. 1–7 (cit. on pp. 76, 127).
- [Byl+22] Zoya Bylinskii, Laura Herman, Aaron Hertzmann, Stefanie Hutka, and Yile Zhang. "Towards Better User Studies in Computer Graphics and Vision". In: *arXiv preprint arXiv:2206.11461* (2022) (cit. on pp. 178, 182).
- [CLA19] Carlos Castillo, Jorge López-Moreno, and Carlos Aliaga. "Recent Advances in Fabric Appearance Reproduction". In: *Computers & Graphics* 84 (2019), pp. 103–121 (cit. on pp. 14, 17, 40, 256, 259).
- [CHB21] Thomas Chambon, Eric Heitz, and Laurent Belcour. "Passing Multi-Channel Material Textures to a 3-Channel Loss". In: *ACM SIGGRAPH 2021 Talks*. 2021, pp. 1–2 (cit. on pp. 64, 66, 67, 128).
- [Cha+19] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. "Everybody Dance Now". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on p. 85).
- [Che+20a] Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. "Towards Learning-based Inverse Subsurface Scattering". In: *2020 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 2020, pp. 1–12 (cit. on p. 203).
- [Che+22] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. "Tensorf: Tensorial Radiance Fields". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 333–350 (cit. on pp. 203, 212).
- [Che+17a] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. "Coherent Online Video Style Transfer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 1105–1114 (cit. on p. 32).
- [Che+17b] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. "Stylebank: An Explicit Representation for Neural Image Style Transfer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1897–1906 (cit. on p. 32).
- [Che+20b] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A Simple Framework for Contrastive Learning of Visual Representations". In: *International Conference on*

- Machine Learning (ICML)*. PMLR. 2020, pp. 1597–1607 (cit. on pp. 37, 45, 159).
- [CXH21] Xinlei Chen, Saining Xie, and Kaiming He. "An Empirical Study of Training Self-supervised Vision Transformers". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2021, pp. 9640–9649 (cit. on p. 159).
- [Che+18] Yuanqin Chen, Qian Zhang, Yaping Wu, Bo Liu, Meiyun Wang, and Yusong Lin. "Fine-tuning ResNet for Breast Cancer Classification from mammography". In: *The International Conference on Healthcare Science and Engineering*. Springer. 2018, pp. 83–96 (cit. on p. 159).
- [CNN22] Zhe Chen, Shohei Nobuhara, and Ko Nishino. "Invertible Neural BRDF for Object Inverse Rendering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12 (2022), pp. 9380–9395 (cit. on pp. 186, 203).
- [Cho+18] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. "Stargan: Unified Generative Adversarial Networks for Multi-domain Image-to-image Translation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8789–8797 (cit. on p. 77).
- [Cla+90] Timothy G Clapp, Hong Peng, Tushar K Ghosh, and Jeffrey W Eischen. "Indirect Measurement of the Moment-curvature Relationship for Fabrics". In: *Textile Research Journal* 60.9 (1990), pp. 525–533 (cit. on pp. 156, 158).
- [Cla+21] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. "Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions". In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 24.3 (2021), pp. 3520–3528 (cit. on pp. 185, 189, 195, 199, 200).
- [CTT17] David Clyde, Joseph Teran, and Rasmus Tamstorf. "Modeling and Data-driven Parameter Estimation for Woven Fabrics". In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2017, pp. 1–11 (cit. on p. 158).
- [Coh+03] Michael F Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. "Wang Tiles for Image and Texture Generation". In: *ACM Transactions on Graphics (TOG)* 22.3 (2003), pp. 287–294 (cit. on p. 72).
- [CT82] Robert L Cook and Kenneth E. Torrance. "A Reflectance Model for Computer Graphics". In: *ACM Transactions on Graphics (TOG)* 1.1 (1982), pp. 7–24 (cit. on p. 48).

- [Dan01] Kristin J Dana. "BRDF/BTF Measurement Device". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Vol. 2. IEEE. 2001, pp. 460–466 (cit. on p. 103).
- [Dan+99] Kristin J Dana, Bram Van Ginneken, Shree K Nayar, and Jan J Koenderink. "Reflectance and Texture of Real-World Surfaces". In: *ACM Transactions On Graphics (TOG)* 18.1 (1999), pp. 1–34 (cit. on p. 33).
- [Dar+12] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. "Image Melding: Combining Inconsistent Images Using Patch-Based Synthesis". In: *ACM Transactions on Graphics (TOG)* 31.4 (2012), pp. 1–10 (cit. on p. 74).
- [Dav+15] Abe Davis, Katherine L Bouman, Justin G Chen, Michael Rubinstein, Fredo Durand, and William T Freeman. "Visual Vibrometry: Estimating Material Properties from Small Motion in Video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5335–5343 (cit. on p. 159).
- [De 97] Jeremy S De Bonet. "Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images". In: *Proceedings of the 24th Annual cConference on Computer Graphics and Interactive Techniques*. 1997, pp. 361–368 (cit. on p. 74).
- [Dek+15] Tali Dekel, Tomer Michaeli, Michal Irani, and William T. Freeman. "Revealing and Modifying Non-Local Variations in a Single Image". In: *ACM Transactions on Graphics (TOG)* (2015) (cit. on p. 93).
- [DH19] Thomas Deliot and Eric Heitz. "Procedural Stochastic Textures by Tiling and Blending". In: *GPU Zen 2* (2019) (cit. on pp. 72, 74, 89–91, 95, 99).
- [Den+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A Large-scale Hierarchical Image Database". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255 (cit. on pp. 32, 164, 166).
- [Des+18] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. "Single-image SVBRDF Capture with a Rendering-Aware Deep Network". In: *ACM Transactions on Graphics (TOG)* 37.4 (2018) (cit. on pp. 17, 34, 39, 54, 77, 102, 123, 125, 127, 258).
- [Des+19] Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. "Flexible SVBRDF Capture

- p>with a Multi-Image Deep Network". In:
- Computer Graphics Forum*
- . Vol. 38. 4. Wiley Online Library. 2019, pp. 1–13 (cit. on pp. 34, 42, 72, 76, 102, 125, 141).
- [DDB20] Valentin Deschaintre, George Drettakis, and Adrien Bousseau. "Guided Fine-Tuning for Large-Scale Material Transfer". In: *Computer Graphics Forum*. Vol. 39. 4. Wiley Online Library. 2020, pp. 91–105 (cit. on pp. 35, 38–40, 47, 49, 50, 63, 72, 76, 105, 125, 142).
- [DLG21] Valentin Deschaintre, Yiming Lin, and Abhijeet Ghosh. "Deep Polarization Imaging for 3D Shape and SVBRDF Acquisition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15567–15576 (cit. on pp. 64, 67, 69, 127, 141).
- [Dia+20] Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. "ResUNet-a: A Deep Learning Framework for Semantic Segmentation of Remotely Sensed Data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162 (2020), pp. 94–114 (cit. on pp. 64, 66, 69, 108, 127, 141).
- [Dia+15] Olga Diamanti, Connelly Barnes, Sylvain Paris, Eli Shechtman, and Olga Sorkine-Hornung. "Synthesis of Complex Image Appearance from Limited Exemplars". In: *ACM Transactions on Graphics (TOG)* 34.2 (2015), pp. 1–14 (cit. on pp. 34, 104).
- [Din+20] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. "Image Quality Assessment: Unifying Structure and Texture Similarity". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020) (cit. on p. 160).
- [DKB14] Laurent Dinh, David Krueger, and Yoshua Bengio. "Nice: Non-linear Independent Components Estimation". In: *arXiv preprint arXiv:1410.8516* (2014) (cit. on p. 191).
- [DSB16] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density Estimation Using Real NVP". In: *arXiv preprint arXiv:1605.08803* (2016) (cit. on p. 191).
- [Dod+22] Ana Dodik, Silvia Sellán, Theodore Kim, and Amanda Phillips. "Sex and Gender in the Computer Graphics Research Literature". In: *arXiv preprint arXiv:2206.00480* (2022) (cit. on pp. 178, 179, 182).
- [DZP07] Weiming Dong, Ning Zhou, and Jean-Claude Paul. "Optimized Tile-Based Texture Synthesis". In: *Proceedings of Graphics Interface 2007*. 2007, pp. 249–256 (cit. on p. 74).
- [Don19] Yue Dong. "Deep Appearance Modeling: A Survey". In: *Visual Informatics* 3.2 (2019), pp. 59–68 (cit. on pp. 34, 53).

- [Don+10] Yue Dong, Jiaping Wang, Xin Tong, John Snyder, Yanxiang Lan, Moshe Ben-Ezra, and Baining Guo. "Manifold Bootstrapping for SVBRDF Capture". In: *ACM Transactions on Graphics (TOG)* 29.4 (2010), pp. 1–10 (cit. on p. 34).
- [DB16] Alexey Dosovitskiy and Thomas Brox. "Generating images with Perceptual Similarity Metrics based on Deep Networks". In: *Advances in Neural Information Processing Systems*. 2016, pp. 658–666 (cit. on p. 74).
- [DTD21] Emilien Dupont, Yee Whye Teh, and Arnaud Doucet. "Generative Models as Distributions of Functions". In: *arXiv preprint arXiv:2102.04776* (2021) (cit. on p. 75).
- [Dur+19a] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. "Neural Spline Flows". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019 (cit. on pp. 185, 191, 192, 194, 196, 197).
- [Dur+19b] Conor Durkan, Artur Bekasovs, Iain Murray, and Georgios Papamakarios. "Cubic-Spline Flows". In: *Workshop on Invertible Neural Nets and Normalizing Flows: ICML 2019*. 2019 (cit. on p. 191).
- [Dye+18] Joanne Dyer, Diego Tamburini, Elisabeth R O'Connell, and Anna Harrison. "A Mul-tispectral Imaging Approach Integrated into the Study of Late Antique textiles from Egypt". In: *PLoS One* 13.10 (2018), e0204699 (cit. on p. 213).
- [EF01] Alexei A Efros and William T Freeman. "Image Quilting for Texture Synthesis and Transfer". In: *Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*. 2001, pp. 341–346 (cit. on pp. 72, 74).
- [EL99] Alexei A Efros and Thomas K Leung. "Texture Synthesis by Non-parametric Sampling". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Vol. 2. IEEE. 1999, pp. 1033–1038 (cit. on p. 74).
- [EM17] Michael Elad and Peyman Milanfar. "Style Transfer via Texture Synthesis". In: *IEEE Transactions on Image Processing* 26.5 (2017), pp. 2338–2351 (cit. on pp. 34, 104).
- [EUD18] Stefan Elfving, Eiji Uchibe, and Kenji Doya. "Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning". In: *Neural Networks* 107 (2018), pp. 3–11 (cit. on pp. 141, 144).
- [End+16] Yuki Endo, Satoshi Iizuka, Yoshihiro Kanamori, and Jun Mitani. "Deepprop: Extracting Deep Features from a Single Image for

- Edit Propagation". In: *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. 2016, pp. 189–201 (cit. on p. 33).
- [Fan+22] Jiahui Fan, Beibei Wang, Miloš Hašan, Jian Yang, and Ling-Qi Yan. "Neural Layered BRDFs". In: *Proceedings of SIGGRAPH 2022*. 2022 (cit. on pp. 116, 187).
- [Fen+22] Xudong Feng, Wenchao Huang, Weiwei Xu, and Huamin Wang. "Learning-Based Bending Stiffness Parameter Estimation by a Drape Tester". In: *ACM Transactions on Graphics (TOG)* 41.6 (2022), pp. 1–16 (cit. on p. 159).
- [Fil+18] J. Filip, M. Kolařová, M. Havlíček, R. Vávra, M. Haindl, and Rushmeier H. "Evaluating Physical and Rendered Material Appearance". In: *The Visual Computer (Computer Graphics International 2018)* (2018) (cit. on p. 109).
- [FH08] Jiří Filip and Michal Haindl. "Bidirectional Texture Function Modeling: A State of the Art Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.11 (2008), pp. 1921–1940 (cit. on p. 103).
- [FR22] Michael Fischer and Tobias Ritschel. "Plateau-Reduced Differentiable Path Tracing". In: *arXiv preprint arXiv:2211.17263* (2022) (cit. on p. 202).
- [Fiš+16] Jakub Fišer, Ondřej Jamříška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Škora. "StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings". In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), pp. 1–11 (cit. on p. 33).
- [Fri+22] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. "Plenoxels: Radiance Fields Without Neural Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5501–5510 (cit. on pp. 187, 212).
- [FAW19] Anna Frühstück, Ibraheem Alhashim, and Peter Wonka. "TileGAN: Synthesis of Large-Scale Non-Homogeneous Textures". In: *ACM Transactions on Graphics (TOG)* 38.4 (Apr. 2019) (cit. on pp. 34, 72, 74, 79–81, 104).
- [Fu+20] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. "Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs". In: *31st British Machine Vision Conference 2020, BMVC 2020*, BMVA Press, 2020 (cit. on pp. 167, 168).
- [GG16] Yarín Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep

- Learning". In: *International Conference on Machine Learning (ICML)*. PMLR. 2016, pp. 1050–1059 (cit. on pp. 123, 126, 130, 141).
- [Gal+12] Bruno Galerne, Ares Lagae, Sylvain Lefebvre, and George Drettakis. "Gabor Noise by Example". In: *ACM Transactions on Graphics (TOG)* 31.4 (2012), pp. 1–9 (cit. on p. 72).
- [Gao+21] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. "Dynamic View Synthesis From Dynamic Monocular Video". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 5712–5721 (cit. on p. 186).
- [Gao+19] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. "Deep Inverse Rendering for High-Resolution SVBRDF Estimation from an Arbitrary Number of Images". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–15 (cit. on pp. 49, 59, 61, 125, 137, 138, 152, 153).
- [GMX22] Duan Gao, Haoyuan Mu, and Kun Xu. "Neural Global Illumination: Interactive Indirect Illumination Prediction under Dynamic Area Lights". In: *IEEE Transactions on Visualization and Computer Graphics* (2022) (cit. on pp. 105, 186, 203).
- [Gar+14] Elena Garces, Aseem Agarwala, Diego Gutierrez, and Aaron Hertzmann. "A Similarity Measure for Illustration Style". In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), pp. 1–9 (cit. on pp. 157, 160, 170, 178, 180).
- [Gar+23] Elena Garces, Victor Arellano, Carlos Rodriguez-Pardo, David Pascual-Hernandez, Sergio Suja, and Jorge Lopez-Moreno. "Towards Material Digitization with a Dual-scale Optical System". In: *ACM Transactions on Graphics (TOG)* (2023) (cit. on pp. 15, 16, 24, 102, 109, 269).
- [Gar+22] Elena Garces, Carlos Rodriguez-Pardo, Dan Casas, and Jorge Lopez-Moreno. "A Survey on Intrinsic Images: Delving Deep into Lambert and Beyond". In: *International Journal of Computer Vision* (2022) (cit. on pp. 3, 4, 6, 7, 12, 22, 23, 127, 128, 141, 268).
- [GES22] James Gardner, Bernhard Egger, and William Alfred Peter Smith. "Rotation-Equivariant Conditional Spherical Neural Fields for Learning a Natural Illumination Prior". In: *Advances in Neural Information Processing Systems*. 2022 (cit. on pp. 19, 185, 186, 193, 196, 199, 202, 203, 260).
- [Gar+19] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. "Deep Parametric Indoor Lighting Estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7175–7183 (cit. on p. 203).

- [GEB15a] Leon Gatys, Alexander S Ecker, and Matthias Bethge. "Texture Synthesis using Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 262–270 (cit. on pp. 74, 95, 128).
- [GEB15b] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "A Neural Algorithm of Artistic Style". In: *arXiv preprint arXiv:1508.06576* (2015) (cit. on pp. 32, 47, 48, 124, 128).
- [GEB16a] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "Image Style Transfer using Convolutional Neural Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2414–2423 (cit. on pp. 74, 160).
- [Gat+17] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. "Controlling Perceptual Factors in Neural Style Transfer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3985–3993 (cit. on pp. 32, 74).
- [GEB16b] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-Decem. June 2016, pp. 2414–2423 (cit. on p. 79).
- [Gau+22] Alban Gauthier, Robin Faury, Jérémy Levallois, Theo Thonat, Jean-Marc Thiery, and Tamy Boubekeur. "MIPNet: Neural Normal-to-Anisotropic-Roughness MIP mapping". In: *ACM Transactions on Graphics (TOG)* 41.6 (2022), pp. 1–12 (cit. on p. 105).
- [Gaw+21] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. "A Survey of Uncertainty in Deep Neural Networks". In: *arXiv preprint arXiv:2107.03342* (2021) (cit. on p. 126).
- [Geo+18] Iliyan Georgiev, Thiago Ize, Mike Farnsworth, Ramón Montoya-Vozmediano, Alan King, Brecht Van Lommel, Angel Jimenez, Oscar Anson, Shinji Ogaki, Eric Johnston, et al. "Arnold: A Brute-force Production Path Tracer". In: *ACM Transactions on Graphics (TOG)* 37.3 (2018), pp. 1–12 (cit. on p. 48).
- [Gil+14] Guillaume Gilet, Basile Sauvage, Kenneth Vanhoey, Jean-Michel Dischler, and Djamchid Ghazanfarpour. "Local Random-phase Noise for Procedural Texturing". In: *ACM Transactions on Graphics (TOG)* 33.6 (2014), pp. 1–11 (cit. on p. 72).

- [Goo+14a] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680 (cit. on p. 39).
- [Goo+14b] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 3. January. Neural information processing systems foundation, June 2014, pp. 2672–2680 (cit. on p. 79).
- [Gri+03] Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, and Peter Schröder. "Discrete Shells". In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Citeseer. 2003, pp. 62–67 (cit. on pp. 161, 162).
- [Gro+20] Aditya Grover, Christopher Chute, Rui Shu, Zhangjie Cao, and Stefano Ermon. "Align-flow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 4028–4035 (cit. on p. 186).
- [Gu+18] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. "Arbitrary Style Transfer with Deep Feature Reshuffle". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8222–8231 (cit. on p. 32).
- [Gua+16] Darya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. "BRDF Representation and Acquisition". In: *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. 2016, pp. 625–650 (cit. on p. 34).
- [Gua+19] Giuseppe Claudio Guarnera, Dar'ya Guarnera, Gregory J Ward, Mashhuda Glencross, and Ian Hall. "BxDF Material Acquisition, Representation, and Rendering for VR and Design". In: *SIGGRAPH Asia 2019 Courses*. 2019, pp. 1–21 (cit. on p. 3).
- [Gue+20] P Guehl, R Allègre, J-M Dischler, B Benes, and E Galin. "Semi-Procedural Textures Using Point Process Texture Basis Functions". In: *Computer Graphics Forum*. Vol. 39. 4. Wiley Online Library. 2020, pp. 159–171 (cit. on pp. 34, 72, 92, 104).
- [Gui+17] Geoffrey Guingo, Basile Sauvage, Jean-Michel Dischler, and Marie-Paule Cani. "Bilayer Textures: A Model for Synthesis and Deformation of Composite Textures". In: *Computer Graphics Forum*. Vol. 36. 4. 2017, pp. 111–122 (cit. on p. 72).

- [Guo+21] Jie Guo, Shuichang Lai, Chengzhi Tao, Yuelong Cai, Lei Wang, Yanwen Guo, and Ling-Qi Yan. "Highlight-Aware Two-Stream Network for Single-Image SVBRDF Acquisition". In: *ACM Transactions on Graphics (TOG)* 40.4 (2021), pp. 1–14 (cit. on pp. 123, 125).
- [Guo+20a] Yu Guo, Miloš Hašan, Lingqi Yan, and Shuang Zhao. "A Bayesian Inference Framework for Procedural Material Parameter Estimation". In: *Computer Graphics Forum*. Vol. 39. 7. Wiley Online Library. 2020, pp. 255–266 (cit. on pp. 125, 126).
- [Guo+20b] Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. "MaterialGAN: Reflectance Capture using a Generative SVBRDF Model". In: *ACM Transactions on Graphics (TOG)* 39.6 (2020), 254:1–254:13 (cit. on pp. 17, 35, 42, 72, 76, 102, 123, 125, 258).
- [Guo+19] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. "Spottune: Transfer Learning through Adaptive Fine-tuning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4805–4814 (cit. on p. 159).
- [HDL16] David Ha, Andrew Dai, and Quoc V Le. "Hypernetworks". In: *arXiv preprint arXiv:1609.09106* (2016) (cit. on p. 75).
- [HMV09] Simon Haegler, Pascal Müller, and Luc Van Gool. "Procedural Modeling for Digital Cultural Heritage". In: *EURASIP Journal on Image and Video Processing* 2009 (2009), pp. 1–1 (cit. on p. 213).
- [Hai12] Vávra R. Haindl M. Filip J. "Digital Material Appearance: the Curse of Tera-Bytes". In: *ERCIM News* 90 (2012), pp. 49–50 (cit. on pp. 108, 109, 111).
- [Ham+21] Hendrik Hameeuw, Godelieve Watteeuw, Bruno Vandermeulen, and Marc Proesmans. "The Painted Panels of the Early Sixteenth Century Mechelen Enclosed Gardens Art Technical Examination with the Photometric Stereo: White Light and Multispectral Microdomes". In: *Papers Presented at the Twentieth Symposium for the Study of Underdrawing and Technology in Painting held in Mechelen and Leuven, 11-13 January 2017*. Vol. 20. Peeters Publishers; Leuven. 2021, pp. 165–175 (cit. on p. 213).
- [Han+22a] Jiyeon Han, Hwanil Choi, Yunjey Choi, Junho Kim, Jung-Woo Ha, and Jaesik Choi. "Rarity Score: A New Metric to Evaluate the Uncommonness of Synthesized Images". In: *arXiv preprint arXiv:2206.08549* (2022) (cit. on p. 128).
- [Han+22b] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing

- Xu, Yixing Xu, et al. "A Survey on Vision Transformer". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022) (cit. on p. 164).
- [Haq+23] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. "Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions". In: *arXiv preprint arXiv:2303.12789* (2023) (cit. on p. 116).
- [HHM22] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. "Shape, Light & Material Decomposition from Images Using Monte Carlo Rendering and Denoising". In: *arXiv preprint arXiv:2206.03380* (2022) (cit. on p. 203).
- [HFM10] Vlastimil Havran, Jiri Filip, and Karol Myszkowski. "Bidirectional Texture Function Compression Based on Multi-level Vector Quantization". In: *Computer Graphics Forum*. Vol. 29. 1. Wiley Online Library. 2010, pp. 175–190 (cit. on p. 103).
- [He+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. "Masked Autoencoders are Scalable Vision Learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 16000–16009 (cit. on p. 159).
- [He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-Decem. 2016, pp. 770–778 (cit. on pp. 64, 66, 69, 78, 94, 108, 109, 127, 141, 164, 176).
- [He+18] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. "Deep Exemplar-Based Colorization". In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), pp. 1–16 (cit. on p. 33).
- [He+19] Mingming He, Jing Liao, Dongdong Chen, Lu Yuan, and Pedro V Sander. "Progressive Color Transfer with Dense Semantic Correspondences". In: *ACM Transactions on Graphics (TOG)* 38.2 (2019), pp. 1–18 (cit. on p. 33).
- [HB95] David J Heeger and James R Bergen. "Pyramid-based Texture Analysis/Synthesis". In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. 1995, pp. 229–238 (cit. on p. 74).
- [Hei20] Eric Heitz. "Can't Invert the CDF? The Triangle-Cut Parameterization of the Region under the Curve". In: *Computer Graphics Forum* 39.4 (2020), pp. 121–132 (cit. on p. 185).
- [HN18] Eric Heitz and Fabrice Neyret. "High-Performance By-Example Noise Using a Histogram-Preserving Blending Operator".

- In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1.2 (2018), pp. 1–25 (cit. on p. 72).
- [Hei+21] Eric Heitz, Kenneth Vanhoey, Thomas Chambon, and Laurent Belcour. "A Sliced Wasserstein Loss for Neural Texture Synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9412–9420 (cit. on pp. 75, 93).
- [Hel+20] Leonhard Helming, Abdelaziz Djelouah, Markus Gross, and Christopher Schroers. "Lossy Image Compression with Normalizing Flows". In: *arXiv preprint arXiv:2008.10486* (2020) (cit. on p. 186).
- [HG16] Dan Hendrycks and Kevin Gimpel. "Gaussian Error Linear Units (GELUs)". In: *arXiv preprint arXiv:1606.08415* (2016) (cit. on pp. 109, 119).
- [Hen+21] Philipp Henzler, Valentin Deschaintre, Niloy J Mitra, and Tobias Ritschel. "Generative Modelling of BRDF Textures from Flash Images". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 40.6 (2021) (cit. on pp. 17, 76, 102, 123, 124, 137, 138, 143, 151–153, 258).
- [Her+20] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. "Deep Geometric Texture Synthesis". In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), pp. 108–1 (cit. on p. 72).
- [Her+01] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. "Image Analogies". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001, pp. 327–340 (cit. on pp. 31, 32, 46, 56, 57, 60).
- [HS05] Aaron Hertzmann and Steven M Seitz. "Example-based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1254–1264 (cit. on p. 34).
- [HS03] Aaron Hertzmann and Steven M Seitz. "Shape and Materials by example: A Photometric Stereo Approach". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. IEEE. 2003, pp. I–I (cit. on p. 34).
- [Heu+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "Gans Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *arXiv preprint arXiv:1706.08500* (2017) (cit. on p. 85).
- [Hil+15] Stephen Hill, Stephen McAuley, Brent Burley, et al. "Physically Based Shading in Theory and Practice". In: *ACM SIGGRAPH*

- 2015 Courses. SIGGRAPH '15. Los Angeles, California: Association for Computing Machinery, 2015 (cit. on p. 11).
- [HS06] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *science* 313.5786 (2006), pp. 504–507 (cit. on p. 103).
- [Hin+21] Tobias Hinz, Matthew Fisher, Oliver Wang, and Stefan Wermter. "Improved Techniques for Training Single-Image GANs". In: *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*. Jan. 2021, pp. 1300–1309 (cit. on p. 75).
- [HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851 (cit. on pp. 127, 209).
- [HS13] Shahera Hossain and Seiichi Serikawa. "Texture Databases—A Comprehensive Survey". In: *Pattern Recognition Letters* 34.15 (2013), pp. 2007–2022 (cit. on p. 109).
- [Hu+20] Bingyang Hu, Jie Guo, Yanjun Chen, Mengtian Li, and Yanwen Guo. "DeepBRDF: A Deep Representation for Manipulating Measured BRDF". In: *Computer Graphics Forum*. Vol. 39. 2. Wiley Online Library. 2020, pp. 157–166 (cit. on pp. 105, 203).
- [Hu+22a] Dongting Hu, Lihua Peng, Tingjin Chu, Xiaoxing Zhang, Yinian Mao, Howard Bondell, and Mingming Gong. "Uncertainty Quantification in Depth Estimation via Constrained Ordinal Regression". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022 (cit. on pp. 126, 130).
- [HSS18] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7132–7141 (cit. on p. 164).
- [HXP20] Wei Hu, Lechao Xiao, and Jeffrey Pennington. "Provable Benefit of Orthogonal Initialization in Optimizing Deep Linear Networks". In: *arXiv preprint arXiv:2001.05992* (2020) (cit. on p. 141).
- [HDR19] Yiwei Hu, Julie Dorsey, and Holly Rushmeier. "A Novel Framework for Inverse Procedural Texture Modeling". In: *ACM Transactions on Graphics (TOG)* 38.6 (2019), pp. 1–14 (cit. on p. 72).
- [Hu+22b] Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. "Controlling Material Appearance by Examples". In: *Computer Graphics Forum*. Vol. 41. 4. Wiley Online Library. 2022, pp. 117–128 (cit. on p. 105).

- [Hu+22c] Yiwei Hu, Chengan He, Valentin Deschaintre, Julie Dorsey, and Holly Rushmeier. "An Inverse Procedural Modeling Pipeline for SVBRDF Maps". In: *ACM Transactions on Graphics (TOG)* 41.2 (2022), pp. 1–17 (cit. on p. 125).
- [Hu+19] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Fredo Durand. "Diff Taichi: Differentiable Programming for Physical Simulation". In: *International Conference on Learning Representations (ICLR)*. 2019 (cit. on p. 158).
- [Hua+18a] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. "Neural Autoregressive Flows". In: *International Conference on Machine Learning (ICML)*. PMLR. 2018, pp. 2078–2087 (cit. on p. 191).
- [HB17] Xun Huang and Serge Belongie. "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 1501–1510 (cit. on p. 32).
- [Hua+18b] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. "Multimodal Unsupervised Image-to-image Translation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018 (cit. on p. 85).
- [Hua+19] Zhiyuan Huang, Mansur Arief, Henry Lam, and Ding Zhao. "Evaluation Uncertainty in Data-Driven Self-Driving Testing". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 1902–1907 (cit. on p. 126).
- [HEW17] Markus Huber, Bernhard Eberhardt, and Daniel Weiskopf. "Cloth Animation Retrieval Using a Motion-shape Signature". In: *IEEE Computer Graphics and Applications* 37.6 (2017), pp. 52–64 (cit. on p. 159).
- [HZ21] Drew A Hudson and C Lawrence Zitnick. "Generative Adversarial Transformers". In: *arXiv preprint arXiv:2103.01209* (2021) (cit. on p. 93).
- [Hwa+22] Inseung Hwang, Daniel S Jeon, Adolfo Muñoz, Diego Gutierrez, Xin Tong, and Min H Kim. "Sparse Ellipsometry: Portable Acquisition of Polarimetric SVBRDF and Shape with Unstructured Flash Photography". In: *ACM Transactions on Graphics (TOG)* 41.4 (2022), pp. 1–14 (cit. on p. 129).
- [Ike81] Katsushi Ikeuchi. "Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1981), pp. 661–669 (cit. on pp. 34, 40, 50, 84, 112, 151).

- [IS15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *International Conference on Machine Learning (ICML)*. PMLR. 2015, pp. 448–456 (cit. on pp. 54, 64, 194).
- [Iso+17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5967–5976 (cit. on pp. 38, 39, 54, 56, 78, 79, 94, 107, 128, 132, 193).
- [Jak+22] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. *Mitsuba 3 Renderer*. Version 3.1.1. <https://mitsuba-renderer.org>. 2022 (cit. on pp. 189, 202).
- [Jam+19] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Šykora. "Stylizing Video by Example". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–11 (cit. on p. 32).
- [Jan+17] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Josh Tenenbaum. "Self-Supervised Intrinsic Image Decomposition". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5936–5946 (cit. on pp. 64, 67, 69, 78).
- [JBH19] Miguel Jaques, Michael Burke, and Timothy Hospedales. "Physics-as-Inverse-Graphics: Unsupervised Physical Parameter Estimation from Video". In: *International Conference on Learning Representations (ICLR)*. 2019 (cit. on p. 158).
- [JCJ09] Wojciech Jarosz, Nathan A. Carr, and Henrik Wann Jensen. "Importance Sampling Spherical Harmonics". In: *Computer Graphics Forum (Proceedings of Eurographics)* 28.2 (2009), pp. 577–586 (cit. on p. 185).
- [JBV16] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. "Texture Synthesis with Spatial Generative Adversarial Networks". In: *arXiv preprint arXiv:1611.08207* (2016) (cit. on pp. 72, 74).
- [Jia+21] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. "Focal Frequency Loss for Image Reconstruction and Synthesis". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 13919–13929 (cit. on pp. 107, 128).
- [Jin+22] Wenhua Jin, Beibei Wang, Miloš Hašan, Yu Guo, Steve Marschner, and Ling-Qi Yan. "Woven Fabric Capture from a

- Single Photo". In: *Proceedings of SIGGRAPH Asia 2022*. 2022 (cit. on p. 125).
- [Jin+19] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. "Neural Style Transfer: A review". In: *IEEE Transactions on Visualization and Computer Graphics* (2019) (cit. on p. 32).
- [JAF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual Losses for Real-time Style Transfer and Super-resolution". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2016, pp. 694–711 (cit. on pp. 32, 74).
- [Jos+22] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. "Hands-on Bayesian Neural NetworksDA Tutorial for Deep Learning Users". In: *IEEE Computational Intelligence Magazine* 17.2 (2022), pp. 29–48 (cit. on p. 126).
- [JC20] Eunjung Ju and Myung Geol Choi. "Estimating Cloth Simulation Parameters From a Static Drape Using Neural Networks". In: *IEEE Access* 8 (2020), pp. 195113–195121 (cit. on pp. 156, 159).
- [KBH03] Florian Kainz, Rod Bogart, and Drew Hess. "The OpenEXR Image File Format". In: *ACM SIGGRAPH Technical Sketches* (2003) (cit. on p. 185).
- [Kaj86] James T Kajiya. "The Rendering Equation". In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 1986, pp. 143–150 (cit. on pp. 4, 8).
- [Kam+16] Christos Kampouris, Stefanos Zafeiriou, Abhijeet Ghosh, and Sotiris Malassiotis. "Fine-Grained Material Classification Using Micro-Geometry and Reflectance". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 778–792 (cit. on p. 40).
- [KG13] Brian Karis and Epic Games. "Real Shading in Unreal Engine 4". In: *Proc. Physically Based Shading Theory Practice 4.3* (2013), p. 1 (cit. on p. 127).
- [Kar+18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations (ICLR)*. 2018 (cit. on pp. 74, 80).
- [Kar+20a] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. "Training Generative Adversarial Networks with Limited Data". In: *arXiv preprint arXiv:2006.06676* (2020) (cit. on pp. 36, 109).

- [Kar+21] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Alias-free Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 852–863 (cit. on p. 186).
- [KLA19] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4401–4410 (cit. on p. 74).
- [Kar+20b] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and Improving the Image Quality of StyleGAN". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 (cit. on pp. 74, 85).
- [Kas+15] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. "Self Tuning Texture Optimization". In: *Computer Graphics Forum*. Vol. 34. 2. 2015, pp. 349–359 (cit. on p. 74).
- [KZP19] Sergey Kastryulin, Dzhamil Zakirov, and Denis Prokopenko. *PyTorch Image Quality: Metrics and Measure for Image Quality Assessment*. Open-source software available at <https://github.com/photosynthesis-team/piq>. 2019 (cit. on pp. 182, 198).
- [Kau17] Eric Kauderer-Abrams. "Quantifying Translation-invariance in Convolutional Neural Networks". In: *arXiv preprint arXiv:1801.01450* (2017) (cit. on p. 37).
- [Kau+00] Jan Kautz, Pere-Pau Vázquez, Wolfgang Heidrich, and Hans-Peter Seidel. "Unified Ap-proach to Prefiltered Environment Maps". In: *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*. 2000, pp. 185–196 (cit. on p. 185).
- [Kaw80] Sueo Kawabata. "The Standardization and Analysis of Hand Evaluation". In: *The Textile Machinery Society of Japan* (1980) (cit. on pp. 156, 158).
- [KG17] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 126).
- [Khe+21] Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. "Causal Autoregressive Flows". In: *International*

- Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3520–3528 (cit. on p. 191).
- [KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. 2015 (cit. on pp. 39, 55, 67, 83, 94, 110, 142, 194).
- [Kin+16] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. "Improved Variational Inference with Inverse Autoregressive Flow". In: *Advances in Neural Information Processing Systems 29* (2016) (cit. on p. 191).
- [KD18] Durk P. Kingma and Prafulla Dhariwal. "Glow: Generative Flow with Invertible 1×1 Convolutions". In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018 (cit. on pp. 186, 191).
- [Kir+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. "Segment Anything". In: *arXiv preprint arXiv:2304.02643* (2023) (cit. on p. 212).
- [KPB20] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2020), pp. 3964–3979 (cit. on p. 191).
- [Kol+20] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. "Big Transfer (bit): General Visual Representation Learning". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 491–507 (cit. on p. 159).
- [KSL19] Simon Kornblith, Jonathon Shlens, and Quoc V Le. "Do Better Imagenet Models Transfer Better?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2661–2671 (cit. on p. 159).
- [Kou+03] Melissa L Koudelka, Sebastian Magda, Peter N Belhumeur, and David J Kriegman. "Acquisition, Compression, and Synthesis of Bidirectional Texture Functions". In: *3rd International Workshop on Texture Analysis and Synthesis (Texture 2003)*. 2003, pp. 59–64 (cit. on p. 103).
- [Kry+21] Michael C Krygier, Tyler LaBonte, Carianne Martinez, Chance Norris, Krish Sharma, Lincoln N Collins, Partha P Mukherjee, and Scott A Roberts. "Quantifying the Unknown Impact of Segmentation Uncertainty on Image-

- Based Simulations". In: *Nature Communications* 12.1 (2021), pp. 1–11 (cit. on p. 126).
- [KLG20] Sandra Kuijpers, Christiane Luible-Bär, and R. Hugh Gong. "The Measurement of Fabric Properties for Virtual Simulation-A Critical Review". In: *IEEE SA INDUSTRY CONNECTIONS* (2020), pp. 1–43 (cit. on p. 158).
- [KL51] Solomon Kullback and Richard A Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86 (cit. on pp. 196, 197).
- [Kum+22] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. "Fine-tuning Can Distort Pretrained Features and Underperform Out-of-distribution". In: *arXiv preprint arXiv:2202.10054* (2022) (cit. on p. 159).
- [Kum+19] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. "Videoflow: A Flow-based Generative Model for Video". In: *arXiv preprint arXiv:1903.01434* 2.5 (2019), p. 3 (cit. on p. 186).
- [Kur+19] Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. "A Large-Scale Study on Regularization and Normalization in GANs". In: *International Conference on Machine Learning (ICML)*. PMLR. 2019, pp. 3581–3590 (cit. on pp. 77, 94).
- [Kur+22] Alexander Kurz, Katja Hauser, Hendrik Alexander Mehrtens, Eva Krieghoff-Henning, Achim Hekler, Jakob Nikolas Kather, Stefan Fröhling, Christof von Kalle, Titus Josef Brinker, et al. "Uncertainty Estimation in Medical Image Classification: Systematic Review". In: *JMIR Medical Informatics* 10.8 (2022) (cit. on p. 126).
- [Kuz+21] Alexandr Kuznetsov, Krishna Mullia, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. "NeuMIP: Multi-Resolution Neural Materials". In: *ACM Transactions on Graphics (TOG)* 40.4 (2021), pp. 1–13 (cit. on pp. 18, 102, 103, 106, 107, 109–116, 119, 120, 187, 207, 259, 264).
- [Kuz+22] Alexandr Kuznetsov, Xuezheng Wang, Krishna Mullia, Fujun Luan, Zexiang Xu, Milos Hasan, and Ravi Ramamoorthi. "Rendering Neural Materials on Curved Surfaces". In: *ACM SIGGRAPH 2022 Conference Proceedings*. 2022, pp. 1–9 (cit. on pp. 102, 104, 116, 187).
- [Kwa+05] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. "Texture Optimization for Example-Based Synthesis". In: *ACM Transactions on Graphics (TOG)*. Vol. 24. 3. ACM. 2005, pp. 795–802 (cit. on pp. 72, 74).

- [Kwa+03] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. "Graphcut Textures: Image and Video Synthesis Using Graph Cuts". In: *ACM Transactions on Graphics (TOG)* 22.3 (2003), pp. 277–286 (cit. on p. 74).
- [LGG19] Manuel Lagunas, Elena Garces, and Diego Gutierrez. "Learning Icons Appearance Similarity". In: *Multimedia Tools and Applications* 78.8 (2019), pp. 10733–10751 (cit. on p. 160).
- [Lag+19] Manuel Lagunas, Sandra Malpica, Ana Serrano, Elena Garces, Diego Gutierrez, and Belen Masia. "A Similarity Measure for Material Appearance". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–12 (cit. on pp. 157, 160, 180).
- [LKA13] Samuli Laine, Tero Karras, and Timo Aila. "Megakernels Considered Harmful: Wavefront Path Tracing on GPUs". In: *Proceedings of the 5th High-Performance Graphics Conference*. 2013, pp. 137–143 (cit. on pp. 109, 195).
- [LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on pp. 126, 130).
- [LS98] Greg Ward Larson and Rob Shakespeare. *Rendering with Radiance: The Art and Science of Lighting Visualization*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998 (cit. on p. 185).
- [Lav+21] Guillaume Lavoué, Nicolas Bonneel, Jean-Philippe Farrugia, and Cyril Soler. "Perceptual Quality of BRDF Approximations: Dataset and Metrics". In: *Computer Graphics Forum*. Vol. 40. 2. Wiley Online Library. 2021, pp. 327–338 (cit. on pp. 130, 131, 193).
- [Led+17] Christian Ledig, Lucas Theis, Ferenc Huszár, et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2017-Janua* (Sept. 2017), pp. 105–114 (cit. on pp. 79, 94).
- [LH06] Sylvain Lefebvre and Hugues Hoppe. "Appearance-Space Texture Synthesis". In: *ACM Transactions on Graphics (TOG)* 25.3 (2006), pp. 541–548 (cit. on pp. 34, 104).
- [Lei+21] Zhao Lei, Yi Zeng, Peng Liu, and Xiaohui Su. "Active Deep Learning for Hyperspectral Image Classification with Uncertainty Learning". In: *IEEE Geoscience and Remote Sensing Letters* 19 (2021) (cit. on p. 126).

- [LM01] Thomas Leung and Jitendra Malik. "Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons". In: *International Journal of Computer Vision* 43.1 (2001), pp. 29–44 (cit. on pp. 33, 39).
- [LLB22] Wei-Hong Li, Xialei Liu, and Hakan Bilen. "Cross-domain Few-shot Learning with Task-specific Adapters". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7161–7170 (cit. on p. 159).
- [Li+17a] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. "Modeling Surface Appearance from a Single Photograph Using Self-Augmented Convolutional Neural Networks". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–11 (cit. on pp. 34, 124).
- [Li+19] Xiaoyu Li, Bo Zhang, Pedro V Sander, and Jing Liao. "Blind Geometric Distortion Correction on Images through Deep Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4855–4864 (cit. on p. 93).
- [Li+22] Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. "DiffCloth: Differentiable Cloth Simulation with Dry Frictional Contact". In: *ACM Transactions on Graphics (TOG)* (2022) (cit. on p. 158).
- [Li+17b] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. "Universal Style Transfer via Feature Transforms". In: *Advances in Neural Information Processing Systems*. 2017, pp. 386–396 (cit. on p. 32).
- [LAA08] Yuanzhen Li, Edward Adelson, and Aseem Agarwala. "ScribbleBoost: Adding Classification to Edge-Aware Interpolation of Local Image and Video Adjustments". In: *Computer Graphics Forum*. Vol. 27. 4. Wiley Online Library. 2008, pp. 1255–1264 (cit. on p. 33).
- [LS18] Zhengqi Li and Noah Snavely. "Learning Intrinsic Image Decomposition From Watching the World". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 (cit. on p. 78).
- [Li+20] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. "Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF From a Single Image". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2475–2484 (cit. on pp. 72, 74, 87–91, 95, 99, 114, 203).

- [LSC18] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. "Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 72–87 (cit. on pp. 72, 125).
- [Li+18] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. "Learning to Reconstruct Shape and Spatially-varying Reflectance from a Single Image". In: *ACM Transactions on Graphics (TOG)* 37.6 (2018), pp. 1–11 (cit. on pp. 125, 129).
- [LLK19] Junbang Liang, Ming Lin, and Vladlen Koltun. "Differentiable Cloth Simulation for Inverse Problems". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 158).
- [Lia+17] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. "Visual Attribute Transfer through Deep Image Analogy". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–15 (cit. on pp. 31, 32, 46, 47, 56, 57, 60).
- [Lic+21] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. "Shape and Material Capture at Home". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 6123–6133 (cit. on p. 129).
- [LPG19] Yiming Lin, Pieter Peers, and Abhijeet Ghosh. "On-Site Example-Based Material Appearance Acquisition". In: *Computer Graphics Forum*. Vol. 38. 4. Wiley Online Library. 2019, pp. 15–25 (cit. on pp. 34, 105).
- [Liu+20] Guilin Liu, Rohan Taori, Ting-Chun Wang, Zhiding Yu, Shiqiu Liu, Fitsum A Reda, Karan Sapra, Andrew Tao, and Bryan Catanzaro. "Transposer: Universal Texture Synthesis Using Feature Maps as Transposed Convolution Filter". In: *arXiv preprint arXiv:2007.07243* (2020) (cit. on pp. 72, 74, 75, 92).
- [Liu+19a] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. "On the Variance of the Adaptive Learning Rate and Beyond". In: *arXiv preprint arXiv:1908.03265* (2019) (cit. on p. 110).
- [Liu+19b] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. "Few-shot Unsupervised Image-to-Image Translation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 10551–10560 (cit. on p. 37).
- [Liu+21] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. "Editing conditional

- radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 5773–5783 (cit. on p. 116).
- [Liu+22a] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. "Neural Rays for Occlusion-aware Image-based Rendering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 7824–7833 (cit. on p. 203).
- [Liu+22b] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. "A Convnet for the 2020s". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11976–11986 (cit. on pp. 108, 109, 119).
- [LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440 (cit. on pp. 83, 94).
- [LH17] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *arXiv preprint arXiv:1711.05101* (2017) (cit. on p. 176).
- [Lua+21] Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. "Unified Shape and SVBRDF Recovery Using Differentiable Monte Carlo Rendering". In: *Computer Graphics Forum*. Vol. 40. 4. Wiley Online Library. 2021, pp. 101–113 (cit. on pp. 125, 129).
- [Lug+20] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. "Srflow: Learning the Super-resolution Space with Normalizing Flow". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 715–732 (cit. on p. 186).
- [LM08] Christiane Luible and Nadia Magnenat-Thalmann. "The Simulation of Cloth Using Accurate Physical Parameters". In: *Proceedings of the Tenth IASTED International Conference on Computer Graphics and Imaging (CGIM '08)*. 2008 (cit. on p. 158).
- [LKS15] Zhaoliang Lun, Evangelos Kalogerakis, and Alla Sheffer. "Elements of Style: Learning Perceptual Shape Style Similarity". In: *ACM Transactions on graphics (TOG)* 34.4 (2015), pp. 1–14 (cit. on p. 160).
- [MHN13] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In:

- International Conference on Machine Learning (ICML)*. Vol. 30. 1. Citeseer. 2013, p. 3 (cit. on pp. 77, 94, 176).
- [Mag+07] Nadia Magnenat-Thalmann, Christiane Luible, Pascal Volino, and Etienne Lyard. "From Measured Fabric to the Simulation of Cloth". In: *2007 10th IEEE International Conference on Computer-Aided Design and Computer Graphics*. IEEE. 2007, pp. 7–18 (cit. on p. 158).
- [MR10] Sébastien Marcel and Yann Rodriguez. "Torchvision the Machine-vision Package of Torch". In: *Proceedings of the 18th ACM International Conference on Multimedia*. 2010, pp. 1485–1488 (cit. on pp. 67, 84, 110, 142, 194).
- [Mar+20] Morteza Mardani, Guilin Liu, Aysegul Dundar, Shiqiu Liu, Andrew Tao, and Bryan Catanzaro. "Neural FFTs for Universal Texture Image Synthesis". In: *Advances in Neural Information Processing Systems* 33 (2020) (cit. on pp. 72, 74, 75, 85, 107, 128).
- [MFS08] Bruce A Maxwell, Richard M Friedhoff, and Casey A Smith. "A bi-illuminant dichromatic reflection model for understanding images". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8 (cit. on p. 10).
- [Maz+19] Ilya Mazlov, Sebastian Merzbach, Elena Trunz, and Reinhard Klein. "Neural Appearance Synthesis and Transfer". In: (2019), pp. 35–39 (cit. on p. 31).
- [MHM18] Leland McInnes, John Healy, and James Melville. "Umap: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv preprint arXiv:1802.03426* (2018) (cit. on p. 133).
- [Meh+21] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. "Modulated Periodic Activations for Generalizable Local Functional Representations". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 14214–14223 (cit. on pp. 109, 186, 203).
- [MR21] Sachin Mehta and Mohammad Rastegari. "MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer". In: *International Conference on Learning Representations (ICLR)*. 2021 (cit. on pp. 124, 127, 132, 141, 144).
- [Mel+12] Francho Melendez, Mashhuda Glencross, Jon Starck, and Gregory J Ward. "Transfer of Albedo and Local Depth Variation to Photo-Textures". In: *Proceedings of the 9th European*

- Conference on Visual Media Production*. 2012, pp. 40–48 (cit. on pp. 31, 34).
- [Mel+21] Joe Mellor, Jack Turner, Amos Storkey, and Elliot J Crowley. "Neural Architecture Search Without Training". In: *International Conference on Machine Learning (ICML)*. PMLR. 2021, pp. 7588–7598 (cit. on p. 92).
- [Mer+19] Sebastian Merzbach, Max Hermann, Martin Rump, and Reinhard Klein. "Learned Fitting of Spatially Varying BRDFs". In: *Computer Graphics Forum*. Vol. 38. 4. Wiley Online Library. 2019, pp. 193–205 (cit. on p. 30).
- [MWK17] Sebastian Merzbach, Michael Weinmann, and Reinhard Klein. "High-quality Multi-Spectral Reflectance Acquisition with X-rite TAC7". In: *Proceedings of the Workshop on Material Appearance Modeling*. 2017, pp. 11–16 (cit. on pp. 35, 39).
- [MNG17] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. "Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks". In: *International Conference on Machine Learning (ICML)*. PMLR. 2017, pp. 2391–2400 (cit. on p. 126).
- [Mic+18] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. "Mixed Precision Training". In: *International Conference on Learning Representations (ICLR)*. 2018 (cit. on pp. 56, 67, 84, 94, 110, 142, 176, 194).
- [Mig+12] Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A Otaduy, and Steve Marschner. "Data-driven Estimation of Cloth Simulation Models". In: *Computer Graphics Forum*. Vol. 31. 2pt2. Wiley Online Library. 2012, pp. 519– 528 (cit. on p. 158).
- [Mil+20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020 (cit. on pp. 186, 187, 207, 212, 262, 264).
- [Min95] Pier Giorgio Minazio. "FAST–Fabric Assurance by Simple Testing". In: *International Journal of Clothing Science and Technology* (1995) (cit. on pp. 156, 158).
- [Miy+18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. "Spectral Normalization for Generative Adversarial

- Networks". In: *International Conference on Learning Representations*. 2018 (cit. on pp. 141, 145).
- [Mor+20] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. "Growing Neural Cellular Automata". In: *Distill* (2020). <https://distill.pub/2020/growing-ca> (cit. on pp. 15, 75, 257).
- [Mor15] Andrew Morgan. *The True Cost*. 2015 (cit. on p. 2).
- [Mor+17] Joep Moritz, Stuart James, Tom S.F. Haines, Tobias Ritschel, and Tim Weyrich. "Texture Stationarization: Turning Photos into Tileable Textures". In: *Eurographics Symposium on Geometry Processing*. Vol. 36. 2. 2017, pp. 177–188 (cit. on pp. 15, 72–74, 88–90, 99, 114, 257).
- [MMK03] Gero Müller, Jan Meseth, and Reinhard Klein. "Compression and Real-Time Rendering of Measured BTFs Using Local PCA." In: *VMV*. 2003, pp. 271–279 (cit. on p. 103).
- [Mül+22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". In: *ACM Transactions on Graphics (TOG)* 41.4 (2022), 102:1–102:15 (cit. on pp. 203, 212).
- [Mül+19] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. "Neural Importance Sampling". In: *ACM Transactions on Graphics (TOG)* 38.5 (2019), pp. 1–19 (cit. on pp. 116, 186, 191, 196, 197).
- [Mül+20] Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. "Neural Control Variates". In: *ACM Transactions on Graphics (TOG)* 39.6 (2020), pp. 1–19 (cit. on pp. 186, 191).
- [Mun+22] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. "Extracting Triangular 3D Models, Materials, and Lighting From Images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 8280–8290 (cit. on pp. 127, 129).
- [Mur+20] J Krishna Murthy, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jérôme Parent-Lévesque, Kevin Xie, Kenny Erleben, et al. "GradSim: Differentiable simulation for System Identification and Visuomotor Control". In: *International Conference on Learning Representations (ICLR)*. 2020 (cit. on p. 158).
- [Naf+16] Hossein Ziaei Nafchi, Atena Shahkolaei, Rachid Hedjam, and Mohamed Cheriet. "Mean Deviation Similarity Index: Efficient

- and Reliable Full-reference Image Quality Evaluator". In: *IEEE Access* 4 (2016), pp. 5579–5590 (cit. on p. 160).
- [Nag+15] Koki Nagano, Graham Fyffe, Oleg Alexander, Jernej Barbic, Hao Li, Abhijeet Ghosh, and Paul E Debevec. "Skin Microstructure Deformation with Displacement Map Convolution." In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 109–1 (cit. on p. 105).
- [NH10] Vinod Nair and Geoffrey E Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *International Conference on Machine Learning (ICML)*. 2010 (cit. on pp. 54, 109, 111, 120).
- [Nam+16] Giljoo Nam, Joo Ho Lee, Hongzhi Wu, Diego Gutierrez, and Min H Kim. "Simultaneous Acquisition of Microscale Reflectance and Normals". In: *ACM Transactions on Graphics (TOG)* 35.6 (2016), pp. 1–11 (cit. on pp. 35, 39).
- [NK18] Hyeonseob Nam and Hyo Eun Kim. "Batch-Instance Normalization for Adaptively Style-Invariant Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 2018-Decem. 2018, pp. 2558–2567 (cit. on p. 77).
- [NSO12] Rahul Narain, Armin Samii, and James F O'brien. "Adaptive Anisotropic Remeshing for Cloth Simulation". In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), pp. 1–10 (cit. on p. 162).
- [ND06] Addy Ngan and Frédo Durand. "Statistical Acquisition of Texture Appearance". In: *Proceedings of the 17th Eurographics Conference on Rendering Techniques*. 2006, pp. 31– 40 (cit. on p. 63).
- [NJR15] Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. "On Optimal, Minimal BRDF Sampling for Reflectance Acquisition". In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), pp. 1–11 (cit. on pp. 42, 116, 130).
- [Nik+21] Eyvind Niklasson, Alexander Mordvintsev, Ettore Randazzo, and Michael Levin. "Self-Organising Textures". In: *Distill* (2021). <https://distill.pub/selforg/2021/textures> (cit. on pp. 72, 74, 75, 89–91, 93, 95, 99).
- [Nim+19] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. "Mitsuba 2: A Retargetable Forward and Inverse Renderer". In: *ACM Transactions on Graphics (TOG)* 38.6 (2019), pp. 1–17 (cit. on pp. 6, 109, 202).
- [ODO16] Augustus Odena, Vincent Dumoulin, and Chris Olah. "Deconvolution and Checker-board Artifacts". In: *Distill* (2016) (cit. on p. 109).

- [Ouy+21] Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. "ReSTIR GI: Path Resampling for Real-time Path Tracing". In: *Computer Graphics Forum*. Vol. 40. 8. 2021, pp. 17–29 (cit. on p. 184).
- [Pap+21] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. "Normalizing Flows for Probabilistic Modeling and Inference". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 2617–2680 (cit. on p. 191).
- [PPM17] George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked Autoregressive Flow for Density Estimation". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 191).
- [PSM19] George Papamakarios, David Sterratt, and Iain Murray. "Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 837–848 (cit. on p. 191).
- [Par+20] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. "Contrastive Learning for Unpaired Image-to-Image Translation". In: *arXiv preprint arXiv:2007.15651* (2020) (cit. on pp. 37, 210).
- [Pas+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An Imperative Style, High-performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019 (cit. on pp. 39, 55, 67, 84, 94, 110, 142, 176, 194).
- [Pha+20] Minh Quang Pham, Josep-Maria Crego, François Yvon, and Jean Senellart. "A study of Residual Adapters for Multi-domain Neural Machine Translation". In: *Conference on Machine Translation*. 2020 (cit. on p. 159).
- [Pha19] Matt Pharr. *Visualizing Warping Strategies for Sampling Environment Map Lights*. 2019 (cit. on pp. 189, 199, 200).
- [PJH20] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Implementation of the forth-coming 4th edition of Physically Based Rendering: From Theory to Implementation*. 2020 (cit. on pp. 3, 109, 189, 200).
- [PJH16] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016 (cit. on pp. 185, 189, 200, 201).

- [PS00] Javier Portilla and Eero P Simoncelli. "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients". In: *International Journal of Computer Vision* 40.1 (2000), pp. 49–70 (cit. on p. 74).
- [Pow13] Jess Power. "Fabric Objective Measurements for Commercial 3D Virtual Garment Simulation". In: *International Journal of Clothing Science and Technology* (2013) (cit. on p. 158).
- [Pra+18] Ekta Prashnani, Hong Cai, Yasamin Mostofi, and Pradeep Sen. "Pieapp: Perceptual Image-error Assessment Through Pairwise Preference". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 1808–1817 (cit. on p. 160).
- [Raa+18] Lara Raad, Axel Davy, Agnès Desolneux, and Jean-Michel Morel. "A Survey of Exemplar-Based Texture Synthesis". In: *Annals of Mathematical Sciences and Applications* 3.1 (2018), pp. 89–148 (cit. on p. 73).
- [Rad+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning Transferable Visual Models from Natural Language Supervision". In: *International Conference on Machine Learning (ICML)*. PMLR. 2021, pp. 8748–8763 (cit. on p. 159).
- [Rag+19] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. "Transfusion: Understanding Transfer Learning for Medical Imaging". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 159).
- [Rai+22] Gilles Rainer, Adrien Bousseau, Tobias Ritschel, and George Drettakis. "Neural Pre-computed Radiance Transfer". In: *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, pp. 365–378 (cit. on pp. 105, 116, 203).
- [Rai+20] Gilles Rainer, Abhijeet Ghosh, Wenzel Jakob, and Tim Weyrich. "Unified Neural Encoding of BTFs". In: *Computer Graphics Forum*. Vol. 39. 2. The Eurographics Association. 2020, pp. 1–13 (cit. on pp. 18, 33, 102, 103, 109, 112, 113, 116, 259).
- [Rai+19] Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. "Neural BTF Compression and Interpolation". In: *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, pp. 235–244 (cit. on pp. 18, 33, 39, 40, 102, 103, 106, 109, 112, 113, 259).
- [Ram+19] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. "Stand-alone Self-attention

- in Vision Models". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 164).
- [RH01a] Ravi Ramamoorthi and Pat Hanrahan. "An Efficient Representation for Irradiance Environment Maps". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, pp. 497–500 (cit. on p. 13).
- [RH01b] Ravi Ramamoorthi and Pat Hanrahan. "An Efficient Representation for Irradiance Environment Maps". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. 2001, pp. 497–500 (cit. on pp. 196, 199, 202).
- [Ras+20] Abdullah Haroon Rasheed, Victor Romero, Florence Bertails-Descoubes, Stefanie Wuhrer, Jean-Sébastien Franco, and Arnaud Lazarus. "Learning to Measure the Static Friction Coefficient in Cloth Contact". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 9912–9921 (cit. on p. 159).
- [RD17] Ajita Rattani and Reza Derakhshani. "On Fine-tuning Convolutional Neural Networks for Smartphone Based Ocular Recognition". In: *2017 IEEE international joint conference on biometrics (IJCB)*. IEEE. 2017, pp. 762–767 (cit. on p. 159).
- [RBV18] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. "Efficient Parametrization of Multi-domain Deep Neural Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8119–8127 (cit. on p. 159).
- [RBV17] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. "Learning Multiple Visual Domains with Residual Adapters". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 159).
- [RJ19] A Sai Bharadwaj Reddy and D Sujitha Juliet. "Transfer Learning with ResNet-50 for Malaria Cell-image Classification". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2019, pp. 0945–0949 (cit. on p. 159).
- [Ree+15] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. "Deep Visual Analogy-Making". In: *Advances in Neural Information Processing Systems*. 2015, pp. 1252–1260 (cit. on p. 32).
- [Rei+14] Klein Reinhard, Rump Martin, Weinmann Michael, Sarlette Ralf, and Schwartz Christopher. "Design and

- Implementation of Practical Bidirectional Texture Function Measurement Devices focusing on the Developments at the University of Bonn". In: *Sensors* 14.5 (2014) (cit. on pp. 109, 114, 117).
- [Rei+18] Rafael Reisenhofer, Sebastian Bosse, Gitta Kutyniok, and Thomas Wiegand. "A Haar Wavelet-based Perceptual Similarity Index for Image Quality Assessment". In: *Signal Processing: Image Communication* 61 (2018), pp. 33–43 (cit. on p. 160).
- [RSS16] Nathalie Remy, Eveline Speelman, and Steven Swartz. *Style that's Sustainable: A New Fast-Fashion Formula*. 2016 (cit. on p. 2).
- [Ren+17] Jian Ren, Xiaohui Shen, Zhe Lin, Radomir Mech, and David J Foran. "Personalized Image Aesthetics". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 638–647 (cit. on p. 74).
- [Rez+20] Danilo Jimenez Rezende, George Papamakarios, Sébastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. "Normalizing Flows on Tori and Spheres". In: *International Conference on Machine Learning (ICML)*. PMLR. 2020, pp. 8083–8092 (cit. on p. 189).
- [Al-+16] Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, et al. "Theano: A Python Framework for Fast Computation of Mathematical Expressions". In: *arXiv e-prints* (2016), arXiv-1605 (cit. on p. 95).
- [Rho+22] Daniel Rho, Junwoo Cho, Jong Hwan Ko, and Eunbyung Park. "Neural Residual Flow Fields for Efficient Video Representations". In: *Proceedings of the Asian Conference on Computer Vision*. 2022, pp. 3447–3463 (cit. on p. 187).
- [Rib+20] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. "Kornia: An Open Source Differentiable Computer Vision Library for Pytorch". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 3674–3683 (cit. on pp. 67, 111, 142, 176).
- [Ric+16] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. "Playing for Data: Ground Truth from Computer Games". In: *European Conference on Computer Vision (ECCV)*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Vol. 9906. LNCS. Springer International Publishing, 2016, pp. 102–118 (cit. on p. 212).

- [RPG16] J  r  my Riviere, Pieter Peers, and Abhijeet Ghosh. "Mobile Surface Reflectometry". In: *Computer Graphics Forum*. Vol. 35. 1. Wiley Online Library. 2016, pp. 191–202 (cit. on pp. 31, 34).
- [Rod+23a] Carlos Rodriguez-Pardo, Henar Dominguez-Elvira, David Pascual-Hernandez, and Elena Garces. "UMat: Uncertainty-Aware Single Image High Resolution Material Capture". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023) (cit. on pp. 24, 62, 76, 102, 107, 108, 110, 116, 193, 268).
- [RG21] Carlos Rodriguez-Pardo and Elena Garces. "Neural Photometry-guided Visual Attribute Transfer". In: *IEEE Transactions on Visualization and Computer Graphics* (2021) (cit. on pp. 23, 25, 63–70, 102, 105, 107, 108, 110, 114, 125, 129, 131, 142, 193, 268).
- [RG22] Carlos Rodriguez-Pardo and Elena Garces. "SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps". In: *IEEE Transactions on Visualization and Computer Graphics* (2022) (cit. on pp. 23, 25, 64, 67, 69, 104, 107, 108, 110, 114, 127, 128, 141, 268).
- [Rod+23b] Carlos Rodriguez-Pardo, Melania Prieto-Martin, Dan Casas, and Elena Garces. "How Will It Drape Like? Capturing Fabric Mechanics from Depth Images". In: *Computer Graphics Forum (Proc. of Eurographics)* (2023) (cit. on pp. 18, 23, 212, 268).
- [Rod+19] Carlos Rodriguez-Pardo, Sergio Suja, David Pascual, Jorge Lopez-Moreno, and Elena Garces. "Automatic Extraction and Synthesis of Regular Repeatable Patterns". In: *Computers & Graphics* 83 (2019), pp. 33–41 (cit. on pp. 15, 34, 72, 75, 79, 83, 87, 89–91, 93, 95, 99, 114, 257).
- [RB19] Carlos Rodriguez-Pardo and Hakan Bilen. "Personalised Aesthetics with Residual Adapters". In: *Iberian Conference on Pattern Recognition and Image Analysis*. Springer. 2019, pp. 508–520 (cit. on pp. 74, 90, 159).
- [Rom+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj  rn Ommer. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10684–10695 (cit. on pp. 127, 209, 210).
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241 (cit. on pp. 37– 39, 54, 55, 64, 66, 67, 69, 78, 108, 111, 124, 127, 132, 141).

- [RSK13] Roland Ruiters, Christopher Schwartz, and Reinhard Klein. "Example-based Interpolation and Synthesis of Bidirectional Texture Functions". In: *Computer Graphics Forum*. Vol. 32. 2pt3. Wiley Online Library. 2013, pp. 361–370 (cit. on p. 63).
- [RSK10] Martin Rump, Ralf Sarlette, and Reinhard Klein. "Groundtruth Data for Multispectral Bidirectional Texture Functions". In: *Conference on Colour in Graphics, Imaging, and Vision*. Vol. 2010. 1. Society for Imaging Science and Technology. 2010, pp. 326–331 (cit. on p. 116).
- [Run+20] Tom FH Runia, Kirill Gavriluk, Cees GM Snoek, and Arnold WM Smeulders. "Cloth in the Wind: A Case Study of Physical Measurement Through Simulation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10498–10507 (cit. on p. 158).
- [Rus+04] Daniel B Russakoff, Carlo Tomasi, Torsten Rohlfing, and Calvin R Maurer. "Image Similarity Using Mutual Information of Regions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2004, pp. 596–607 (cit. on p. 132).
- [Sah+22] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. "Palette: Image-to-Image Diffusion Models". In: *ACM SIGGRAPH 2022 Conference Proceedings*. 2022, pp. 1–10 (cit. on pp. 127, 209).
- [San+19] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. "Data Augmentation Using Generative Adversarial Networks (CycleGAN) to Improve Generalizability in CT Segmentation Tasks". In: *Scientific reports* 9.1 (2019), pp. 1–9 (cit. on p. 45).
- [SC20] Shen Sang and Manmohan Chandraker. "Single-Shot Neural Relighting and SVBRDF Estimation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 85–101 (cit. on p. 125).
- [SOC22] Igor Santesteban, Miguel A Otaduy, and Dan Casas. "SNUG: Self-Supervised Neural Dynamic Garments". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022) (cit. on pp. 17, 259).
- [SSK03] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. "Efficient and Realistic Visualization of Cloth". In: *Rendering techniques*. 2003, pp. 167–178 (cit. on p. 109).
- [SSK20] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. "A U-Net Based Discriminator for Generative Adversarial Networks". In:

- Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 8207–8216 (cit. on pp. 83, 93, 124, 128, 132, 141, 142).
- [Sch+17] Vincent Schüssler, Eric Heitz, Johannes Hanika, and Carsten Dachsbacher. "Microfacet-based Normal Mapping for Robust Monte Carlo Path Tracing". In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), pp. 1–12 (cit. on p. 201).
- [See66] Robert T Seeley. "Spherical Harmonics". In: *The American Mathematical Monthly* 73.4P2 (1966), pp. 115–121 (cit. on pp. 185, 196).
- [Sel+20] Raghavendra Selvan, Frederik Faye, Jon Middleton, and Akshay Pai. "Uncertainty Quantification in Medical Image Segmentation with Normalizing Flows". In: *Machine Learning in Medical Imaging*. 2020, pp. 80–90 (cit. on p. 186).
- [SKK18] Murat Sensoy, Lance Kaplan, and Melih Kandemir. "Evidential Deep Learning to Quantify Classification Uncertainty". In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on pp. 126, 130).
- [SDM19] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. "SinGAN: Learning a Generative Model From a Single Natural Image". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on pp. 32, 38, 57, 75, 85, 86, 90).
- [Shi+20] Liang Shi, Beichen Li, Miloš Hašan, Kalyan Sunkavalli, Tamy Boubekour, Radomir Mech, and Wojciech Matusik. "Match: Differentiable Material Graphs for Procedural Material Capture". In: *ACM Transactions on Graphics (TOG)* (2020) (cit. on pp. 76, 125, 137, 138, 143, 151–153).
- [Sho+19] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. "InGAN: Capturing and Retargeting the "DNA" of a Natural Image". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on p. 75).
- [SCI18] Assaf Shocher, Nadav Cohen, and Michal Irani. "'Zero-Shot' Super-Resolution Using Deep Internal Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 (cit. on p. 75).
- [SK19] Connor Shorten and Taghi M Khoshgoftaar. "A Survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6.1 (2019), p. 60 (cit. on pp. 36, 45).
- [SZ15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition".

- In: *International Conference on Learning Representations (ICLR)*. 2015 (cit. on pp. 32, 164).
- [Sin+21] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. "Negative Data Augmentation". In: *arXiv preprint arXiv:2102.05113* (2021) (cit. on p. 93).
- [Sit+20] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. "Implicit Neural Representations with Periodic Activation Functions". In: *Advances in Neural Information Processing Systems* 33 (2020) (cit. on pp. 75, 102, 109, 112, 120, 186, 193, 194, 262).
- [Sit+21] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. "Light Field Networks: Neural Scene Representations with Single-Evaluation Rendering". In: *Advances in Neural Information Processing Systems*. 2021 (cit. on p. 187).
- [SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments". In: *ACM Transactions on Graphics (TOG)* 21.3 (July 2002), pp. 527–536 (cit. on p. 13).
- [Sne17] Xavier Snelgrove. "High-resolution Multi-scale Neural Texture Synthesis". In: *SIGGRAPH Asia 2017 Technical Briefs*. 2017, pp. 1–4 (cit. on p. 74).
- [Sol+21] Ava P Soleimany, Alexander Amini, Samuel Goldman, Daniela Rus, Sangeeta N Bhatia, and Connor W Coley. "Evidential Deep Learning for Guided Molecular Property Prediction and Discovery". In: *ACS Central Science* 7.8 (2021), pp. 1356–1367 (cit. on pp. 126, 135).
- [Spe+22] Georg Sperl, Rosa M Sánchez-Banderas, Manwen Li, Chris Wojtan, and Miguel A Otaduy. "Estimation of Yarn-level Simulation Models for Production Fabrics". In: *ACM Transactions on Graphics (TOG)* 41.4 (2022), pp. 1–15 (cit. on p. 163).
- [Sri+20] Pratul P Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T Barron, Richard Tucker, and Noah Snavely. "Lighthouse: Predicting Lighting Volumes for Spatially-Coherent Illumination". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 8080–8089 (cit. on p. 203).
- [Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way

- to Prevent Neural Networks from Overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958 (cit. on pp. 110, 127, 141, 176).
- [Ste+14] Heinz Christian Steinhausen, Dennis den Brok, Matthias B. Hullin, and Reinhard Klein. "Acquiring Bidirectional Texture Functions for Large-Scale Material Samples". In: (2014) (cit. on pp. 30, 63).
- [Ste+15a] Heinz Christian Steinhausen, Dennis den Brok, Matthias B. Hullin, and Reinhard Klein. "Extrapolating Large-Scale Material BTFs under Cross-Device Constraints". In: *Vision, Modeling & Visualization*. Ed. by David Bommes, Tobias Ritschel, and Thomas Schultz. The Eurographics Association, 2015, pp. 143–150 (cit. on pp. 33, 104).
- [Ste+15b] Heinz Christian Steinhausen, Rodrigo Martín, Dennis den Brok, Matthias B. Hullin, and Reinhard Klein. "Extrapolation of Bidirectional Texture Functions using Texture Synthesis guided by Photometric Normals". In: *Measuring, Modeling, and Reproducing Material Appearance II (SPIE 9398)*. Vol. 9398. 14. San Francisco, USA, Feb. 2015 (cit. on pp. 33, 104).
- [Str+22] Yannick Strümler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. "Implicit Neural Representations for Image Compression". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 74–91 (cit. on p. 187).
- [SGK21] Vadim Sushko, Juergen Gall, and Anna Khoreva. "One-Shot GAN: Learning to Generate Samples from Single Images and Videos". In: *arXiv preprint arXiv:2103.13389* (2021) (cit. on p. 75).
- [SB08] Cédric Syllebranque and Samuel Boivin. "Estimation of Mechanical Parameters of Deformable Solids from Videos". In: *The Visual Computer* 24.11 (2008), pp. 963–972 (cit. on p. 158).
- [Szt+21] Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. "Neural BRDF Representation and Importance Sampling". In: *Computer Graphics Forum*. Vol. 40. 6. Wiley Online Library. 2021, pp. 332–346 (cit. on pp. 105, 111, 116, 186, 187, 193, 203, 210).
- [Tak+21] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. "Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cit. on p. 186).

- [Tam+11] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. "Adaptively Learning the Crowd Kernel". In: *International Conference on Machine Learning (ICML)*. 2011, pp. 673–680 (cit. on pp. 170, 171).
- [Tan+22] Matthew Tancik, Vincent Casser, Xincheng Yan, Sabeek Pradhan, Ben Mildenhall, Pratul Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. "Block-NeRF: Scalable Large Scene Neural View Synthesis". In: *arXiv*. 2022 (cit. on p. 187).
- [Tan+20] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains". In: *Advances in Neural Information Processing Systems* (2020) (cit. on pp. 75, 186, 193).
- [Tew+22] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. "Advances in Neural Rendering". In: *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, pp. 703–735 (cit. on pp. 187, 212).
- [Tex+20a] Ondřej Texler, David Futschik, Jakub Fišer, Michal Lukáč, Jingwan Lu, Eli Shechtman, and Daniel Šykora. "Arbitrary Style Transfer using Neurally-Guided Patch-Based Ssynthesis". In: *Computers & Graphics* 87 (2020), pp. 62–71 (cit. on pp. 39, 45).
- [Tex+20b] Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Mencelei Chai, Sergey Tulyakov, and Daniel Šykora. "Interactive Video Stylization Using Few-Shot Patch-Based Training". In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), pp. 73–1 (cit. on pp. 30, 31, 33, 37, 39, 46, 53, 58, 61, 107, 108, 128, 129).
- [TOB15] Lucas Theis, Aäron van den Oord, and Matthias Bethge. "A Note on the Evaluation of Generative Models". In: *arXiv preprint arXiv:1511.01844* (2015) (cit. on p. 41).
- [Tom94] Shoji Tominaga. "Dichromatic reflection models for a variety of materials". In: *Color Research & Application* 19.4 (1994), pp. 277–285 (cit. on p. 10).
- [Ton+02] Xin Tong, Jingdan Zhang, Ligang Liu, Xi Wang, Baining Guo, and Heung-Yeung Shum. "Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces". In: *ACM Transactions on Graphics (TOG)* 21.3 (2002), pp. 665–672 (cit. on p. 103).

- [TS67] Kenneth E Torrance and Ephraim M Sparrow. "Theory for Off-Specular Reflection from Roughened Surfaces". In: *Josa* 57.9 (1967), pp. 1105–1114 (cit. on p. 11).
- [Tu+20] Peihan Tu, Li-Yi Wei, Koji Yatani, Takeo Igarashi, and Matthias Zwicker. "Continuous Curve Textures". In: *ACM Transactions on Graphics (TOG)* 39.6 (2020), pp. 1–16 (cit. on p. 72).
- [TK74] Amos Tversky and Daniel Kahneman. "Judgment under Uncertainty: Heuristics and Biases: Biases in Judgments Reveal Some Heuristics of Thinking Under Uncertainty." In: *Science* 185.4157 (1974), pp. 1124–1131 (cit. on p. 169).
- [Uly+16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. "Texture Networks: Feed-Forward Synthesis of Textures and Stylized Images." In: *International Conference of Machine Learning (ICML)*. Vol. 1. 2. 2016, p. 4 (cit. on p. 76).
- [UVL18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep Image Prior". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 (cit. on p. 75).
- [UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Instance Normalization: The Missing Ingredient for Fast Stylization". In: *arXiv preprint arXiv:1607.08022* (2016) (cit. on pp. 77, 94).
- [Van11] Dietger G Van Antwerpen. "Unbiased Physically Nased Rendering on the GPU". MA thesis. Electrical Engineering, Mathematics and Computer Science, 2011 (cit. on p. 195).
- [Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, "ukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 164).
- [VG95] Eric Veach and Leonidas J. Guibas. "Optimally Combining Sampling Techniques for Monte Carlo Rendering". In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. Association for Computing Ma-chinery, 1995 (cit. on p. 184).
- [VPS21] Giuseppe Vecchio, Simone Palazzo, and Concetto Spampinato. "SurfaceNet: Adversarial SVBRDF Estimation from a Single Image". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 12840–12848 (cit. on pp. 17, 76, 123, 125, 128, 129, 132, 258).
- [Ver+22] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. "Ref-NeRF:

- Structured View-Dependent Appearance for Neural Radiance Fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2022 (cit. on pp. 187, 212).
- [VZH20] Yael Vinker, Nir Zabari, and Yedid Hoshen. "Training End-to-end Single Image Generators without GANs". In: *arXiv preprint arXiv:2004.06014* (2020) (cit. on p. 75).
- [VMF09] Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. "A Simple Approach to Nonlinear Tensile Stiffness for Accurate Cloth Simulation". In: *ACM Transactions on Graphics (TOG)* 28.4 (2009), Article–No (cit. on pp. 158, 161, 162).
- [Wal+14] Ingo Wald, Sven Woop, Carsten Benthin, Gregory S. Johnson, and Manfred Ernst. "Embree: A Kernel Framework for Efficient CPU Ray Tracing". In: *ACM Transactions on Graphics (TOG)* 33.4 (2014) (cit. on p. 195).
- [Wal+07] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. "Microfacet Models for Refraction through Rough Surfaces". In: *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. 2007, pp. 195–206 (cit. on p. 127).
- [WZY22] Cairong Wang, Yiming Zhu, and Chun Yuan. "Diverse Image Inpainting with Normal-izing Flow". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 53–69 (cit. on p. 186).
- [Wan+22a] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. "Clip-nerf: Text-and-image Driven Manipulation of Neural Radiance Fields". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 3835–3844 (cit. on p. 116).
- [Wan+22b] Chen Wang, Xiang Wang, Jiawei Zhang, Liang Zhang, Xiao Bai, Xin Ning, Jun Zhou, and Edwin Hancock. "Uncertainty Estimation for Stereo Matching Based on Evidential Deep Learning". In: *Pattern Recognition* 124 (2022), p. 108498 (cit. on p. 130).
- [Wan+18] Guotai Wang, Wenqi Li, Maria A Zuluaga, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. "Interactive Medical Image Segmentation using Deep Learning with Image-specific Fine Tuning". In: *IEEE transactions on medical imaging* 37.7 (2018), pp. 1562–1573 (cit. on p. 159).
- [WOR11] Huamin Wang, James F O'Brien, and Ravi Ramamoorthi. "Data-driven Elastic Models for Cloth: Modeling and Measurement".

- In: *ACM Transactions on Graphics (TOG)* 30.4 (2011), pp. 1–12 (cit. on p. 158).
- [Wan+09] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. "All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance". In: 28.5 (Dec. 2009), pp. 1–10 (cit. on p. 13).
- [Wan+22c] Jiayi Wang, Diogo Luvizon, Franziska Mueller, Florian Bernard, Adam Kortylewski, Dan Casas, and Christian Theobalt. "HandFlow: Quantifying View-Dependent 3D Ambiguity in Two-Hand Reconstruction with Normalizing Flow". In: *International Symposium on Vision, Modeling, and Visualization*. 2022 (cit. on p. 186).
- [Wan+20a] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X. Yu. "Orthogonal Con-volutional Neural Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 (cit. on p. 93).
- [Wan+22d] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. "Fourier Plenotrees for Dynamic Radiance Field Rendering in Real-time". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 13524–13534 (cit. on pp. 203, 212).
- [Wan+20b] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. "Linformer: Self-attention with Linear Complexity". In: *arXiv preprint arXiv:2006.04768* (2020) (cit. on pp. 124, 127, 141, 144).
- [Wan+19] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. "Few-shot Video-to-Video Synthesis". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5013–5024 (cit. on p. 37).
- [Wan+20c] Xiaofei Wang, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey". In: *IEEE Communications Surveys & Tutorials* 22.2 (2020), pp. 869–904 (cit. on p. 212).
- [Wan+20d] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. "Generalizing From a Few Examples: A Survey on Few-Shot Learning". In: *ACM Computing Surveys (CSUR)* 53.3 (2020), pp. 1–34 (cit. on p. 37).
- [Wan+04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Transactions on Image*

- Processing* 13.4 (2004), pp. 600–612 (cit. on pp. 50, 54, 85, 86, 90, 92, 160, 198, 199, 202).
- [Wan+21] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. "Learning Indoor Inverse Rendering with 3D Spatially-Varying Lighting". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 12538–12547 (cit. on p. 203).
- [Wan+23] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jonathan Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. "Neural Fields meet Explicit Geometric Representations for Inverse Rendering of Urban Scenes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023 (cit. on pp. 203, 212).
- [WGK14] Michael Weinmann, Juergen Gall, and Reinhard Klein. "Material Classification Based on Training Data Synthesized Using a BTF Database". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer International Publishing, 2014, pp. 156–171 (cit. on pp. 50, 103, 108, 111, 151).
- [WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A Survey of Transfer Learning". In: *Journal of Big data* 3.1 (2016), pp. 1–40 (cit. on p. 159).
- [Wen+22] Tao Wen, Beibei Wang, Lei Zhang, Jie Guo, and Nicolas Holzschuch. "SVBRDF Recovery from a Single Image with Highlights Using a Pre-trained Generative Adversarial Network". In: *Computer Graphics Forum*. Wiley Online Library. 2022 (cit. on p. 125).
- [Woo+18] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. "Cbam: Convolutional block attention module". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 3–19 (cit. on pp. 109, 119, 128, 141, 145, 164).
- [WTX22] Qiling Wu, Jianchao Tan, and Kun Xu. "PaletteNeRF: Palette-based Color Editing for NeRFs". In: *arXiv preprint arXiv:2212.12871* (2022) (cit. on p. 116).
- [WH18] Yuxin Wu and Kaiming He. "Group Normalization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 3–19 (cit. on pp. 64, 67, 69, 127, 141, 144).
- [Xie+22] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. "Neural Fields in Visual Computing and Beyond". In: *Computer Graphics Forum* (2022) (cit. on pp. 187, 212).

- [Xu+13] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. "Anisotropic Spherical Gaussians". In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), pp. 1–11 (cit. on pp. 185, 196, 199, 202).
- [Yan+21] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. "Geometry processing with Neural Fields". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 22483–22497 (cit. on p. 186).
- [YLL17] Shan Yang, Junbang Liang, and Ming C Lin. "Learning-based Cloth Material Recovery from Video". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 4383–4393 (cit. on pp. 156, 159).
- [YL15] Shan Yang and Ming C Lin. "Materialcloning: Acquiring Elasticity Parameters from Images for Medical Applications". In: *IEEE Transactions on Visualization and Computer Graphics* 22.9 (2015), pp. 2122–2135 (cit. on p. 158).
- [Yan+18] Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C Lin. "Physics-inspired Garment Recovery from a Single-view Image". In: *ACM Transactions on Graphics (TOG)* 37.5 (2018), pp. 1–14 (cit. on p. 158).
- [Yao+22] Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. "NeILF: Neural Incident Light Field for Material and Lighting Estimation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022 (cit. on p. 187).
- [Ye+22] Weicai Ye, Shuo Chen, Chong Bao, Hujun Bao, Marc Pollefeys, Zhaopeng Cui, and Guofeng Zhang. "Intrinsicnerf: Learning Intrinsic Neural Radiance Fields for Editable Novel View Synthesis". In: *arXiv preprint arXiv:2210.00647* (2022) (cit. on p. 116).
- [Ye+21] Wenjie Ye, Yue Dong, Pieter Peers, and Baining Guo. "Deep Reflectance Scanning: Recovering Spatially-varying Material Appearance from a Flash-lit Video Sequence". In: *Computer Graphics Forum*. Vol. 40. 6. Wiley Online Library. 2021, pp. 409–427 (cit. on p. 125).
- [Ye+18] Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. "Single Image Surface Appearance Modeling with Self-Augmented CNNs and Inexact Supervision". In: *Computer Graphics Forum*. Vol. 37. 7. Wiley Online Library. 2018, pp. 201–211 (cit. on pp. 34, 125).
- [Yep20] Tyler Yep. *torchinfo*. Mar. 2020 (cit. on p. 113).

- [YS19] Ye Yu and William AP Smith. "InverseRenderNet: Learning Single Image Inverse Rendering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3155–3164 (cit. on pp. 78, 203).
- [YS21] Ye Yu and William AP Smith. "Outdoor Inverse Rendering from a Single Image using Multiview Self-supervision". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2021), pp. 3659–3675 (cit. on p. 203).
- [Yun+19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. "Cutmix: Regularization strategy to train strong classifiers with localizable features". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6023–6032 (cit. on p. 93).
- [Zha+19a] Bo Zhang, Mingming He, Jing Liao, Pedro V Sander, Lu Yuan, Amine Bermak, and Dong Chen. "Deep Exemplar-based Video Colorization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8052–8061 (cit. on p. 33).
- [Zha+19b] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. "Self-attention Generative Adversarial Networks". In: *International Conference on Machine Learning (ICML)*. PMLR. 2019, pp. 7354–7363 (cit. on pp. 164, 166, 176).
- [ZDN16] Hang Zhang, Kristin Dana, and Ko Nishino. "Friction from Reflectance: Deep Re-reflectance Codes for Predicting Physical Surface Properties from One-shot In-field Reflectance". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 808–824 (cit. on p. 159).
- [Zha+20] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. "Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity". In: *International Conference on Learning Representations (ICLR)*. 2020 (cit. on pp. 110, 194).
- [Zha+21] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. "Physg: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 5453–5462 (cit. on p. 203).
- [ZL12] Lin Zhang and Hongyu Li. "SR-SIM: A Fast and High Performance IQA Index Based on Spectral Residual". In: *IEEE International Conference on Image Processing*. IEEE. 2012, pp. 1473–1476 (cit. on p. 160).

- [ZSL14] Lin Zhang, Ying Shen, and Hongyu Li. "VSI: A Visual Saliency-induced Index for Perceptual Image Quality Assessment". In: *IEEE Transactions on Image processing* 23.10 (2014), pp. 4270–4281 (cit. on p. 160).
- [Zha+11] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. "FSIM: A Feature Similarity Index for Image Quality Assessment". In: *IEEE Transactions on Image Processing* 20.8 (2011), pp. 2378–2386 (cit. on p. 160).
- [Zha+18] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 586–595 (cit. on pp. 32, 39, 41, 49, 50, 54, 57, 67, 85, 86, 90, 110, 128, 142, 157, 160, 169, 170, 178, 198, 199, 202).
- [ZJK20] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. "Exploring Self-attention for Image Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10076–10085 (cit. on p. 164).
- [Zho+20] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. "Random Erasing Data Augmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2020 (cit. on pp. 111, 129, 142, 176).
- [Zho+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learn-ing Deep Features for Discriminative Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2921–2929 (cit. on p. 164).
- [Zho+22] Xilong Zhou, Miloš Hašan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Kalantari. "TileGen: Tileable, Controllable Material Generation and Capture". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* (2022) (cit. on pp. 105, 107, 109, 116, 123).
- [Zho+23] Xilong Zhou, Miloš Hašan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Khademi Kalantari. "A Semi-Procedural Convolutional Material Prior". In: *Computer Graphics Forum*. Wiley Online Library. 2023 (cit. on p. 105).
- [ZK21] Xilong Zhou and Nima Khademi Kalantari. "Adversarial Single-Image SVBRDF Estimation with Hybrid Training". In: *Computer Graphics Forum*. Vol. 40. 2. Wiley Online Library. 2021, pp. 315–325 (cit. on pp. 123, 125, 127, 137, 138, 141, 152, 153).
- [Zho+17] Yang Zhou, Huajie Shi, Dani Lischinski, Minglun Gong, Johannes Kopf, and Hui Huang. "Analysis and Controlled

- Synthesis of Inhomogeneous Textures". In: *Computer Graphics Forum*. Vol. 36. 2. 2017, pp. 199–212 (cit. on p. 74).
- [Zho+18] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. "Non-Stationary Texture Synthesis by Adversarial Expansion". In: *ACM Transactions on Graphics (TOG)* 37.4 (July 2018) (cit. on pp. 34, 38, 56, 72, 74, 76, 77, 79, 84, 85, 87, 90, 94, 104).
- [Zho+19] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. "Context-Reinforced Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4046–4055 (cit. on p. 41).
- [Zhu+17a] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* 2017-Octob (Mar. 2017), pp. 2242–2251 (cit. on pp. 78, 79, 83, 94).
- [Zhu+17b] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. "Toward Multimodal Image-to-Image Translation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 465–476 (cit. on p. 37).
- [Zhu+21] Wei Zhu, Xian Guo, Dai Owaki, Kyo Kutsuzawa, and Mitsuhiro Hayashibe. "A Survey of Sim-to-real Transfer Techniques Applied to Reinforcement Learning for Bioinspired Robots". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021) (cit. on pp. 156, 212).

A. RESUMEN

Las escenas virtuales hiperrealistas están cada vez más presentes en nuestra sociedad, con una amplia variedad de aplicaciones en áreas como la fabricación, la arquitectura, el diseño de moda y el entretenimiento, incluyendo películas, videojuegos y realidad aumentada y virtual. Generar imágenes realistas de tales escenas requiere modelos de iluminación, geometría y materiales altamente precisos, lo que puede ser costoso de obtener en recursos y tiempo. Tradicionalmente, estos modelos a menudo han sido creados manualmente por artistas entrenados, pero este proceso puede ser prohibitivamente ineficiente. Alternativamente, se pueden capturar ejemplos del mundo real, pero este enfoque presenta desafíos adicionales en términos de precisión y escalabilidad. Además, aunque el realismo y la precisión son cruciales en esos procesos, la eficiencia en el renderizado también es un requisito clave, para que se puedan generar imágenes realistas con la velocidad requerida en muchas aplicaciones del mundo real. Uno de los desafíos más significativos en este sentido es la adquisición y representación de materiales, que son un componente crítico de nuestro mundo visual y, por extensión, de las representaciones virtuales del mismo. Sin embargo, los enfoques existentes para la adquisición y representación de materiales están limitados en términos de eficiencia y precisión, lo que restringe su impacto en el mundo real. Para abordar estos desafíos, los enfoques basados en datos que aprovechan el aprendizaje automático pueden proporcionar soluciones viables. Sin embargo, el diseño y entrenamiento de modelos de redes neuronales que satisfagan todos estos requisitos sigue siendo un desafío, que requiere una cuidadosa consideración de los compromisos entre calidad y eficiencia.

En esta tesis, proponemos soluciones novedosas basadas en aprendizaje automático para abordar varios desafíos clave en la renderización de escenas virtuales. Nuestros métodos hacen uso de diversas formas de redes neuronales para introducir algoritmos innovadores para la codificación de la radiancia, la generación digital de materiales, su edición y estimación. En primer lugar, presentamos un sistema de transferencia de atributos visuales que puede generalizar eficazmente a nuevas condiciones de iluminación y distorsiones geométricas, que ampliamos para la adquisición de materiales de alta resolución mediante un dispositivo diseñado para tal propósito. También, proponemos un modelo generativo capaz de sintetizar texturas teselables – que se puede concatenar consigo misma sin que aparezcan artefactos de repetición – a partir de una sola imagen de entrada, lo que ayuda a mejorar la calidad de la renderización de materiales. Esta tesis también presenta una representación de materiales que codifica con precisión la reflectancia del material y ofrece potentes capacidades de edición y propagación. Para lograr este objetivo, nos basamos en trabajos recientes en campos neuronales. Además de la reflectancia, presentamos un método novedoso para la codificación de iluminación

global que aprovecha modelos generativos cuidadosamente diseñados para lograr un muestreo significativamente más rápido y de mayor calidad que los trabajos disponibles en la literatura. Por último, proponemos dos métodos innovadores para la digitalización escalable de materiales. Usando escáneres como dispositivo de captura, presentamos un modelo generativo que puede proporcionar estimaciones de reflectancia de materiales de alta resolución utilizando una sola imagen como entrada. Adicionalmente, introducimos el primer algoritmo de cuantificación de incertidumbre que aumenta su fiabilidad y eficiencia. Además, presentamos un método neuronal para digitalizar las propiedades mecánicas de tejidos utilizando imágenes de profundidad como entrada, que extendemos con una métrica de similitud perceptualmente validada mediante un estudio de usuario. Globalmente, las contribuciones de esta tesis representan avances significativos en los campos de la codificación de la radiancia y la adquisición y edición digital de materiales, lo que permite mejorar la calidad, escalabilidad y eficiencia de los sistemas de renderización.

A.1 Antecedentes

Generación Digital de Materiales

La digitalización precisa y de alta resolución de la reflectancia de los materiales es un problema crucial para generar entornos virtuales atractivos y realistas. Este proceso de digitalización se realiza típicamente utilizando un *goniorrefleatómetro*, una máquina altamente compleja diseñada para medir la respuesta fotométrica de un material utilizando fuentes de luz y cámaras colocadas en diferentes posiciones alrededor de su semiesfera. Los dispositivos existentes para medir la apariencia del material en muestras espacialmente variables se limitan a una sola escala, ya sea micro o mesoscópica. Esta es una limitación práctica cuando el material tiene una estructura compleja a múltiples escalas, como es común en muchos materiales del mundo real. Para ellos, medir con precisión su reflectancia espacialmente variable requiere imágenes microscópicas de alta resolución. Por ejemplo, el comportamiento óptico de los textiles depende mucho de sus fibras o microestructuras [CLA19]. Además, muchos materiales también presentan variaciones *mesoscópicas* a gran escala, como patrones de impresión o los tartanes, que no se pueden capturar utilizando únicamente fotografías microscópicas. Además, muchos de estos dispositivos no pueden capturar propiedades ópticas importantes, como la transmitancia del material, que es clave para la representación realista de objetos como los tejidos. No hay dispositivos de captura ni algoritmos disponibles para la digitalización precisa y de alta resolución de *Spatially Varying Bidirectional Scattering Distribution Function* (SVBSDF) para materiales heterogéneos. Para introducir tales materiales, que son muy comunes en muchas industrias como la moda, la textil o la fabricación de cuero, en los sistemas de representación, es necesario

crear estas máquinas y desarrollar algoritmos precisos para manejar los datos que capturan. Finalmente, dicho dispositivo puede servir como fuente de generación de datos para entrenar modelos que operen en escenarios de datos más limitados, como en la digitalización a partir de una sola imagen.

Edición y Síntesis

SVBRDFs Teselables Incluso si la reflectancia puede ser medida con precisión para una gran superficie de un material, en entornos de renderización, también es importante que el material digital sea *teselable*. Una textura teselable es una imagen de textura que puede ser repetida infinitamente en todas las direcciones sin presentar discontinuidades visibles. En otras palabras, una textura teselable está diseñada de tal manera que se conecta perfectamente con copias idénticas de sí misma, que pueden ser dispuestas una al lado de la otra o apiladas una encima de la otra para crear un patrón más grande y continuo sin ninguna repetición visible. Esto se utiliza a menudo en gráficos por ordenador, especialmente para crear texturas en modelos 3D, o incluso para la colocación de fondos en sitios web. En los últimos años, se han propuesto diferentes métodos para la generación de texturas teselables a partir de un solo ejemplo de entrada [Mor+17; Rod+19; Mor+20]; sin embargo, estos trabajos presentan varias limitaciones para la generación de *Spatially Varying Bidirectional Reflectance Distribution Functions* (SVBRDFs) teselables. En primer lugar, todos suponen un cierto nivel de regularidad en las imágenes de entrada, o bien las texturas generadas pierden una cantidad significativa de fidelidad visual. Además, estos métodos se limitan a generar un solo mapa de textura (normalmente, el albedo del material). Esto es problemático para la síntesis de SVBRDFs teselables, que requieren transformar cada mapa de textura en el SVBRDF en un mapa teselable, al mismo tiempo que se preserva la coherencia píxel a píxel entre los mapas. En resumen, en la actualidad no es posible generar SVBRDFs teselables de alta calidad a partir de un solo ejemplo de entrada con los métodos propuestos en trabajos previos.

Transferencia de Atributos Visuales Las representaciones efectivas de materiales requieren comprender las propiedades que los definen de manera única, a las que nos referimos como *atributos visuales*. Estos atributos son parámetros que varían espacialmente y mantienen la coherencia espacial con respecto a la estructura del material, mientras que permanecen in-variantes ante los cambios en la iluminación de la escena o la geometría del objeto subyacente. Por ejemplo, pueden representar propiedades ópticas de un SVBRDF, estilizaciones artísticas o máscaras de segmentación semántica. Obtener estos atributos para muestras grandes de materiales es un problema que presenta numerosas dificultades. Una forma de abordarlo es obtener estos atributos en un pequeño ejemplo del material y *transferirlos* a porciones más grandes del mismo, utilizando una imagen de entrada grande que sirve como guía. En los últimos años, se han propuesto diferentes métodos

de transferencia de atributos visuales, sin embargo, están limitados en su eficiencia, robustez, generalización o rendimiento. Idealmente, un método de transferencia de atributos visuales para materiales debería ser robusto ante nuevas condiciones de iluminación del material, degradaciones de la cámara o distorsiones de entrada; además de ser eficiente (proporcionando tiempos de transferencia interactivos incluso en presupuestos computacionales bajos), controlable y predecible, y capaz de generar salidas de alta resolución, que son necesarias para SVBRDFs realistas. Desarrollar un método con estas características puede resultar útil para muchas tareas posteriores, como sistemas eficientes de generación de datos o digitalización de materiales.

Digitalización Escalable

Estimación de la Reflectancia de Materiales Como ya hemos mencionado, la digitalización precisa de los materiales típicamente requiere dispositivos costosos que utilizan cantidades significativas de tiempo e intervención manual para operar. Estos dispositivos proporcionan representaciones digitales altamente realistas de la reflectancia del material a costa de la escalabilidad, lo que limita su aplicabilidad en muchos entornos del mundo real. Por ejemplo, muchos procesos de fabricación, como los de la industria textil, generan una amplia variedad de diferentes materiales que no pueden ser digitalizados fácil ni económicamente con la cadencia requerida en estas industrias. Además, la disponibilidad limitada de estos dispositivos introduce ineficiencias adicionales y costes económicos y ecológicos, ya que las muestras físicas de los materiales deben ser enviadas desde el cliente a la localización del dispositivo de captura.

La estimación de la reflectancia de materiales a partir de dispositivos de bajo coste es un problema que se ha estudiado ampliamente en la literatura [AWL13; AWL15; Des+18; Hen+21; Guo+20b; VPS21]. Estos métodos prometen proporcionar una digitalización de materiales escalable, eficiente y económica. Normalmente, se basan en una o varias imágenes de un material iluminadas con flash tomadas con una cámara de smartphone, y en un modelo de aprendizaje profundo entrenado en materiales sintéticos que proporciona estimaciones plausibles. Sin embargo, estos enfoques presentan varias limitaciones que los hacen inadecuados para flujos de trabajo en los que se requiere digitalización precisa. En primer lugar, muchos de estos métodos se basan en modelos generativos que tienden a producir artefactos o utilizan grandes redes neuronales que limitan la resolución de salida, lo que reduce la calidad de los materiales estimados. Además, las imágenes tomadas con smartphones presentan importantes desafíos de calibración, mientras que la mayoría de los conjuntos de datos existentes utilizados para entrenar estos modelos son puramente sintéticos, lo que limita aún más la calidad de estas estimaciones. Además, muchos de estos métodos utilizan funciones de coste perceptuales para el entrenamiento o la optimización en tiempo de evaluación,

lo que, si bien ayuda a digitalizar materiales estocásticos, presenta desafíos adicionales para garantizar la repetibilidad y consistencia necesarias para construir un inventario digital. Además de estas limitaciones, la estimación de reflectancia de materiales a partir de una sola imagen sigue siendo un problema mal planteado y, por lo tanto, muchos resultados diferentes pueden ser plausibles dados los mismos datos de entrada. Hasta ahora, no existe una forma de cuantificar la incertidumbre para este problema, lo que dificulta la aplicabilidad de estos sistemas en escenarios del mundo real, y limita su fiabilidad y eficiencia.

Comportamiento Mecánico de Materiales Textiles Además de la reflectancia de los materiales, otro componente importante de las escenas virtuales es la geometría de los objetos. Mientras que muchos objetos del mundo real tienen formas estáticas, que pueden ser escaneadas con dispositivos relativamente accesibles como el sensor *LiDAR* en *iPhones*, existen materiales que cambian de forma cuando se les aplican fuerzas externas, como la gravedad. En este sentido, los materiales deformables requieren modelos computacionales y sistemas de captura específicos. Uno de los materiales más comunes de este tipo son las telas, que son ubicuas en el mundo real y se utilizan en nuestra ropa, muebles o vehículos. Para ellas, capturar con precisión su forma estática no es suficiente: Es necesario poder simular su comportamiento en nuevos escenarios, como nuevas prendas o fuerzas dinámicas, como el viento [Ber+23], o el movimiento humano [SOC22]. Además, las telas son increíblemente variadas en sus patrones de fabricación (por ejemplo, diferentes patrones de tejido o estructuras de punto), composiciones y acabados, lo que no solo determina su apariencia óptica [CLA19] sino también su comportamiento mecánico. El problema de capturar las propiedades mecánicas de las telas para que puedan ser re-simuladas en entornos virtuales ha sido ampliamente estudiado en la literatura; sin embargo, las soluciones actuales requieren una intervención humana tediosa o dispositivos de captura costosos, lo que dificulta su escalabilidad. Además, existe una falta de entendimiento sobre la similitud perceptual del comportamiento mecánico de las telas, lo que crea desafíos adicionales para diseñar sistemas precisos de estimación de parámetros mecánicos de tejidos.

Codificación Neuronal de Radiancia

Reflectancia Las *Bidirectional Texture Functions (BTFs)* proporcionan representaciones densas y de gran exactitud de la reflectancia de los materiales, sin embargo, son típicamente prohibitivas en términos computacionales y de memoria. Trabajo reciente [Rai+19; Rai+20; Kuz+21] presenta representaciones neuronales para BTFs, consiguiendo resultados de muy alta calidad y eficiencia. Los métodos neuronales de codificación de reflectancia entrenados en BTF sintéticos o medidos han mostrado resultados prometedores para aumentar el realismo de los materiales renderizados. Sin embargo, estas

codificaciones son inmutables, lo que significa que su salida para una consulta determinada de UV, cámara y vector de luz queda fija una vez que se entrenan. Esto puede ser muy limitante cuando el fragmento del material utilizado para el entrenamiento es demasiado pequeño o no se puede teselar, lo que ocurre con frecuencia cuando el material ha sido escaneado con un dispositivo de captura. Por lo tanto, las codificaciones de reflectancia neuronal actuales, aunque precisas y eficientes, tienen importantes limitaciones en cuanto a las capacidades de edición, lo que dificulta su aplicabilidad y utilidad en entornos de renderizado del mundo real.

Iluminación Existen numerosas aproximaciones para la representación de iluminación en escenas virtuales, incluyendo *Spherical Harmonics*, *Spherical Gaussians* o *Haar Wavelets*. Estas, sin embargo, tienen dificultades para representar de manera eficiente y precisa condiciones de iluminación no difusas, como fuentes de luz de alta frecuencia, como el sol o las bombillas. Trabajo reciente sobre representaciones neuronales de la iluminación natural [GES22] proporciona mejores aproximaciones para estos casos, pero tiene dificultades con la iluminación interior o las escenas nocturnas con múltiples fuentes de luz. Además, estas aproximaciones neuronales no proporcionan capacidades de muestreo y evaluación de PDF, que son esenciales para el renderizado de Monte Carlo en el trazado de rayos. No existen métodos de aproximación de iluminación que proporcionen estas capacidades de muestreo, a la vez que funcionen con precisión y eficiencia en cualquier tipo de mapa de entorno de entrada. Trabajo reciente sobre modelos generativos invertibles y representaciones neuronales implícitas puede proporcionar un camino prometedor para resolver estos problemas.

A.2 Objetivos

El objetivo principal de esta tesis es crear algoritmos innovadores para la representación computacional y la captura de materiales y la radiancia de escenas. Esto es importante por varios motivos. Estas representaciones digitales pueden ser utilizadas en una amplia variedad de aplicaciones y problemas, incluyendo gráficos por ordenador, videojuegos, producción de películas, realidad virtual y aumentada, o diseño computacional. Tienen el potencial de aumentar significativamente el realismo visual y la calidad de los entornos virtuales creados en estos sistemas. Además, la precisión y la eficiencia de estos métodos tienen un impacto directo en la calidad y los recursos computacionales requeridos para estas tareas. Por lo tanto, obtener representaciones y algoritmos de digitalización precisos y eficientes para materiales y otros componentes de las escenas virtuales puede ayudar a reducir el costo computacional de la renderización, el diseño, la captura y la modelización, lo que puede llevar a sistemas más rápidos,

eficientes y escalables. Además, pueden permitir la creación procedural eficiente de conjuntos de datos para entrenar modelos para resolver otras tareas, como segmentación o clasificación, o generación de imágenes o videos. Finalmente, además de la eficiencia y la precisión, características adicionales como la diferenciabilidad o la capacidad de cuantificación de la incertidumbre crean oportunidades para aplicaciones de optimización o el aprendizaje activo.

En esta tesis, presentamos nuevos métodos basados en aprendizaje automático para abordar los problemas descritos anteriormente. Nuestras soluciones propuestas buscan lograr la máxima eficiencia computacional y de datos, así como la precisión, el control y la fiabilidad. Además, nuestros algoritmos están diseñados para ser beneficiosos para los usuarios finales al incorporar requisitos humanos como componentes perceptuales, de edición, predictibilidad, robustez y cuantificación de incertidumbre. Todos nuestros métodos propuestos aprovechan las capacidades de las redes neuronales, incorporando los últimos avances en representaciones implícitas, diseño de arquitectura y funciones de coste, y procedimientos de entrenamiento. Todos nuestros modelos son completamente diferenciables y se pueden incorporar en los procesos de optimización para resolver problemas de renderización inversa. Nuestras soluciones para la codificación de la radiancia se pueden incorporar sin problemas en motores de trazado de rayos para aumentar la eficiencia del renderizado o en representaciones neuronales de escenas para aumentar el realismo. Además, nuestros métodos de digitalización casual de alta calidad permiten la captura escalable de materiales, lo que puede ayudar a los usuarios finales a crear sus propios materiales virtuales realistas o crear grandes conjuntos de datos a un menor coste.

Los objetivos de esta tesis se pueden resumir en tres puntos:

- Diseñar nuevos métodos basados en algoritmos de aprendizaje automático para la generación, propagación, edición y síntesis de materiales digitales, con altos estándares de predecibilidad, control y calidad.
- Introducir nuevas representaciones para la radiancia de escenas, que sean eficientes, editables y diferenciables.
- Democratizar la captura digital de alta calidad de materiales mediante el diseño de nuevos métodos de bajo coste, escalables, fiables y validados perceptualmente.

A.3 Metodología

En esta sección detallamos la metodología seguida durante el transcurso de esta tesis.

Revisión Bibliográfica

Una vez definidos los objetivos de la tesis, descritos anteriormente, se revisó la bibliografía de trabajos previos para identificar huecos en la literatura y encontrar líneas de investigación prometedoras. En este proceso de revisión de la bibliografía, encontramos una importante ausencia de métodos capaces de generar datos de alta calidad para el entrenamiento de modelos de captura y representación de materiales; además de la oportunidad de entrenar modelos de captura y edición de materiales más precisos, eficientes y robustos. Es importante recalcar que esta revisión se realizó durante todas las etapas de la tesis, ya que los campos del aprendizaje automático, la informática gráfica y la visión por ordenador se mueven a una velocidad muy elevada. En ellos, nuevas soluciones innovadoras se proponen a diario, por lo que mantener vivo este análisis del estado del arte es fundamental para el desarrollo efectivo de algoritmos. Por ejemplo, nuevas tendencias como los campos neuronales [Mil+20] y las representaciones implícitas [Sit+20] han sido identificadas y han sido claves para el desarrollo de muchos de los algoritmos propuestos en esta tesis.

Generación de Materiales Digitales

Los primeros proyectos desarrollados en el contexto de esta tesis están enfocados en la generación de materiales digitales de alta calidad, los cuales pueden ser utilizados posteriormente para otras tareas, como entrenar representaciones neuronales de materiales o crear grandes conjuntos de datos para modelos de estimación de materiales. Estos métodos toman como entrada un solo material, lo que ayuda a aumentar la predictibilidad de sus resultados.

En el capítulo 2, presentamos un método neuronal para la de propagación de atributos visuales capaz de transferir, para un material dado, muchos tipos de mapas de propiedades visuales a imágenes de nuevos parches del mismo material, o uno similar. Estas nuevas imágenes pueden ser capturadas bajo cualquier condición de iluminación, variaciones de color, distorsiones geométricas o configuración de cámara. Este modelo utiliza redes neuronales convolucionales de imagen a imagen con pocos parámetros, para proporcionar la primera solución capaz de aprovechar el comportamiento óptico del material para el problema de transferencia de atributos visuales, al ser entrenado con un conjunto de datos fotométrico y una política amplia de aumentado sintético de datos. Este método puede ser entrenado en menos de un minuto y es capaz de transferir cualquier tipo de atributo visual, desde las normales de la superficie, hasta estilizaciones artísticas o mapas de segmentación semántica. Sin embargo, está limitado en precisión, sólo puede aprender de un conjunto de datos a la vez y sólo puede aprender a transferir un atributo visual.

Para abordar estas limitaciones, en el capítulo 2.C ampliamos este método, presentando una red neuronal más compleja capaz de generar mapas de SVBSDF de gran calidad y alta resolución para cualquier material, dados uno o varios conjuntos de datos fotométricos que representan pequeñas porciones del material. A través de extensas modificaciones en el procedimiento de entrenamiento y mejoras en la arquitectura del modelo, no sólo habilitamos esta transferencia de atributos visuales con la capacidad de aprender de múltiples conjuntos de datos y transferir varias imágenes simultáneamente, sino que también aumentamos la precisión y nitidez de sus estimaciones. Sin embargo, estas mejoras incrementan el coste computacional del entrenamiento de la red neuronal y limitan su generalidad. Con estos modelos, podemos construir materiales a gran escala, que pueden ser utilizados para generar datos BTF para entrenar otras representaciones (Capítulo 4), o construir grandes conjuntos de datos para entrenar modelos que se generalicen a nuevos materiales (Capítulo 5).

Las texturas utilizadas en aplicaciones de renderización, además de representar una gran parte de las variaciones del material, necesitan ser teselables. Es decir, cuando se concatenan espacialmente consigo mismas, no debe ser posible identificar estas repeticiones. En el capítulo 3, presentamos una Red Generativa Antagónica (GAN) entrenada con una sola imagen, que puede generar múltiples texturas teselables del mismo material. Para lograr esto, ampliamos trabajos anteriores sobre síntesis de texturas para permitir texturas teselables mediante un nuevo algoritmo de manipulación de espacios latentes; y para manejar múltiples mapas de textura simultáneamente, mediante modificaciones en la arquitectura neuronal. Además, aprovechamos el discriminador de la GAN como función de estimación de calidad en tiempo de evaluación, lo que ayuda a distinguir entre generaciones de alta y baja calidad. Este modelo logra texturas teselables de mayor calidad que trabajos anteriores, en tiempos computacionales más bajos.

Estos modelos dependen de un solo material como entrada, lo que, si bien aumenta la pre-dictibilidad, limita sus capacidades de generalización. No obstante, estos métodos proporcionan evidencia de que, bajo ciertas suposiciones y mediante procedimientos de entrenamiento altamente controlados, las redes neuronales pueden ser utilizadas para generar datos de alta calidad. Estos datos pueden ser aprovechados para otras tareas, como se demuestra en otras secciones de esta tesis.

Codificación Neuronal de Radiancia

Uno de los problemas de investigación que exploramos en esta tesis es la representación de radiancia utilizando redes neuronales, lo que ofrece varios beneficios potenciales en comparación con enfoques tradicionales. Las redes

neuronales pueden proporcionar funciones eficientes, continuas y completamente diferenciables que son útiles para la renderización directa y para problemas de renderización inversa. Además, pueden permitir novedosas capacidades de edición o muestreo eficiente para la renderización mediante Monte Carlo. En este trabajo, nos centramos en dos componentes clave de las escenas virtuales: los materiales y la iluminación. Nuestro objetivo es desarrollar enfoques basados en redes neuronales para codificar estos elementos que sean precisos y computacionalmente eficientes, abriendo el camino para nuevas aplicaciones en informática gráfica y visión por ordenador.

Respecto a los materiales en el capítulo 4 presentamos una nueva representación neuronal para codificar su reflectancia. Construyendo sobre trabajos previos sobre materiales neuronales [Kuz+21], nuestro método tiene tres componentes: Una *textura neuronal*, que codifica la reflectancia del material en un espacio latente 2D; un *renderizador neuronal* completamente convolucional y con activaciones sinusoidales, que toma como entrada esta reflectancia latente y los ángulos de luz y visión y genera valores lineales RGB; y un *autoencoder*, que puede estimar la textura neuronal de cualquier imagen de entrada 2D del material. Al igual que en trabajos previos sobre campos neuronales [Mil+20], entrenamos cada modelo desde cero para cada material, tomando como entrada su *Bidirectional Texture Function* (BTF), que puede ser capturada o generada sintéticamente. Nuestro enfoque introduce la primera representación neuronal para BTFs con entrada condicional, lo que hace posible la extrapolación de BTFs, la creación de BTF teselables y la síntesis de nuevos materiales mediante propagación de reflectancia. Abordamos una de las principales limitaciones de los trabajos previos sobre materiales neuronales, que era su falta de capacidades de edición, lo que impedía su aplicabilidad a escenarios de renderización del mundo real que requieren materiales de gran escala y teselables. A pesar de estas mejoras, nuestra representación tiene limitaciones. Por ejemplo, propiedades semánticas como la especularidad o el albedo se codifican en el espacio latente y, por tanto, no son fácilmente editables. Sin embargo, creemos que nuestro método tiene un potencial significativo para trabajos futuros sobre representaciones neuronales de materiales.

En el capítulo 7, abordamos el problema de la codificación de la iluminación, presentando un nuevo método neuronal para el muestreo, la codificación de la PDF y la compresión RGB de los mapas de entorno utilizados para la iluminación global en la renderización. Nuestro método se basa en trabajos anteriores sobre *representaciones neuronales implícitas* para comprimir el mapa de entorno HDRI en una función continua y diferenciable. Diseñamos un *normalizing flow* que puede muestrear direcciones de mapas de entorno y medir su densidad de probabilidad. Ambas funcionalidades son esenciales para el muestreo por importancia en la renderización mediante Monte Carlo. Nuestro diseño de arquitectura neuronal y de sus procedimientos de entrenamiento nos permiten obtener modelos que son hasta dos órdenes

de magnitud más rápidos que los métodos analíticos. Utilizando un conjunto de datos de mapas de entorno con propiedades diversas, demostramos alta generalidad y precisión, superando trabajos anteriores en este campo. Para fomentar futuras investigaciones, proporcionaremos una implementación de código abierto y un conjunto de datos de modelos entrenados. Si bien este es el primer método que aprende a muestrear de mapas de entorno, creemos que existe potencial para mejorar aún más nuestro enfoque mediante la introducción de diseños de redes neuronales más sofisticados o aprendiendo distribuciones a priori sobre la iluminación global.

Digitalización Escalable de Materiales

Uno de los objetivos clave de esta tesis es desarrollar soluciones de digitalización que sean escalables y asequibles, sin depender de hardware costoso o inaccesible. Para lograrlo, hemos realizado dos contribuciones diferentes. En primer lugar, en el capítulo 5, presentamos un método novedoso para estimar las *Spatially-Varying Bidirectional Reflectance Distribution Functions* (SVBRDF) utilizando escáneres como dispositivo de captura. Este método se basa en gran medida en los algoritmos de generación de datos presentados en secciones anteriores de esta tesis. Nuestro enfoque utiliza una red neuronal ligera con mecanismos de atención, lo que resulta en estimaciones con una resolución considerablemente mayor que los métodos anteriores que se utilizaban imágenes tomadas con smartphones como entrada. Observamos empíricamente que la reflectancia del material depende en gran medida de su microgeometría, que los escáneres planos pueden capturar adecuadamente, a la vez que proporcionan imágenes que se pueden utilizar directamente como el *albedo* de la SVBRDF. Una de las contribuciones clave de este trabajo es la introducción del primer algoritmo de cuantificación de incertidumbre para la estimación de materiales, para el cual demostramos aplicaciones en la creación de conjuntos de datos a través del aprendizaje activo. Además, una extensión de este trabajo forma parte de [Textura.ai](https://textura.ai), que es utilizada por usuarios reales, lo que demuestra su robustez y fiabilidad. Tenemos confianza en que las novedades introducidas por este método darán forma a futuros avances sobre el problema de la digitalización de materiales a partir de una sola imagen. Aunque confiamos en las capacidades de nuestro método, este se puede extender para estimar propiedades de reflectancia adicionales como la anisotropía o la transmitancia, para generalizarse a otros materiales o dispositivos, y para mejorar aún más su exactitud.

Además, en el capítulo 6, proponemos un método para la estimación de parámetros mecánicos de materiales textiles utilizando imágenes de profundidad y la densidad del material como entradas. Nuestro método logra una mayor escalabilidad que los trabajos anteriores, que requerían secuencias de vídeo, intervención manual o dispositivos costosos y específicos. Para lograrlo, proponemos

una solución que es agnóstica a la apariencia óptica del tejido y que puede ser utilizada por operadores no expertos. Utilizamos una red neuronal que predice el conjunto completo de parámetros mecánicos de cualquier tejido utilizando datos sintéticos para su entrenamiento, aprovechando una política extensiva de aumento de datos, aprendizaje por transferencia, mecanismos de atención y múltiples formas de agrupamiento de activaciones internas. Una contribución significativa de este trabajo es nuestra propuesta de una métrica de similitud de propiedades mecánicas en tejidos, que mide las distancias entre el comportamiento mecánico de los tejidos utilizando solo imágenes como entrada. Validamos esta métrica con un estudio de usuarios, demostrando que se relaciona en gran medida con la percepción humana. Aunque nuestro método tiene contribuciones significativas para la estimación del comportamiento mecánico de los textiles, todavía está limitado en al menos dos formas: utiliza imágenes de profundidad como entrada, que son relativamente fáciles de obtener pero menos accesibles que las imágenes RGB, y la métrica propuesta de similitud no es fácilmente diferenciable, lo que dificulta su potencial como función de pérdida en problemas de optimización.

A.4 Resultados

Estas son las principales contribuciones de esta tesis:

- Un método basado en aprendizaje profundo para la propagación de atributos visuales de un material a muestras más grandes del mismo material, o de un material similar. Usando una red neuronal convolucional con un número reducido de parámetros, en-trenada mediante un *dataset fotométrico* y una política extensiva de aumentado de datos, el modelo puede generalizar a nuevas condiciones de iluminación, cámara, color o distorsiones geométricas. Enseñamos las capacidades de nuestro método para la transferencia de atributos visuales de varios tipos, como las normales de la superficie del material, mapas de segmentación semántica o ediciones artísticas. (Capítulo 2).
- Una extensión al método anterior, para habilitar un sistema de captura a dos escalas, que permite digitalizar un material a altos niveles de resolución y precisión. Para ello, extendemos el método presentado en el capítulo 2 para permitir la transferencia de varios mapas de propiedades simultáneamente, con un mayor nivel de exactitud. (Capítulo 2.C).
- Un modelo generativo capaz de generar *texturas teselables* usando una sola imagen como entrada. Explotando las capacidades de las redes generativas antagónicas para la síntesis de texturas, un innovador algoritmo de manipulación de espacios latentes, y usando el discriminador como una métrica de calidad, nuestro modelo es capaz de generar texturas teselables a un mayor nivel de calidad perceptual y a un coste computacional que métodos anteriores. Además, proponemos una extensión a este método para la generación de SVBRDFs teselables usando un sólo material como entrada,

- y enseñamos las capacidades de nuestro algoritmo para sintetizar texturas de diferentes niveles de regularidad. (Capítulo 3).
- Una representación basada en campos neuronales capaz de codificar y transferir BTFs. Nuestro modelo se construye sobre trabajo anterior en representaciones neuronales de materiales, y proponemos una representación latente ligera que puede ser descodificada por una red neuronal implícita en valores de reflectancia por téxel. Esta representación latente es estimada por una red neuronal del tipo *autoencoder*, que puede ser utilizada para propagar valores de BTF a estructuras nuevas, habilitando nuevas capacidades de edición y transferencia de materiales neuronales, proporcionando nuevas competencias a este tipo de representaciones. (Capítulo 4).
 - Un innovador sistema de digitalización de materiales capaz de, a partir de una sola imagen de entrada, generar mapas de SVBRDF a una alta resolución, usando un escáner como dispositivo de captura. Nuestro sistema usa un conjunto de datos generado para este propósito y un novedoso modelo de translación de imagen a imagen con mecanismos de atención entrenado con funciones de coste específicamente diseñadas para este problema. Este modelo es capaz de generar materiales digitales con un alto nivel de exactitud y detalle, usando la microgeometría del material como la principal señal de supervisión. Además, proponemos el primer algoritmo para la cuantificación de incertidumbre en problemas de digitalización de materiales, construyendo sobre métodos de aprendizaje profundo Bayesiano, *dropout* de Monte Carlo y métricas perceptuales de BRDF. Demostramos que nuestra métrica es eficaz para la predicción de error en digitalización en el momento de la evaluación, y para la generación eficiente de datasets mediante aprendizaje activo. (Capítulo 5).
 - Un novedoso sistema de digitalización de propiedades mecánicas de tejidos usando un sistema casual de captura e imágenes de profundidad. Entrenamos nuestro modelo usando únicamente datos sintéticos, y lo diseñamos para hacer uso de mecanismos de atención neuronales, aprendizaje por transferencia, y diferentes mecanismos de agregación de características latentes, además de una extensiva política de aumentado de datos. Nuestro modelo es capaz de predecir con exactitud las propiedades mecánicas de fragmentos de materiales textiles sin la necesidad de costosos sistemas de captura. Además, proporcionamos una innovadora métrica perceptual para medir la similitud entre materiales textiles en función de sus parámetros mecánicos, la cual validamos mediante un estudio de usuario. (Capítulo 6).
 - Un método neuronal para mapas de entorno, que habilita una representación eficiente para iluminación global en escenas virtuales. Para ello, usamos dos redes neuronales diferentes: Un *normalizing flow*, capaz de aprender la distribución de probabilidad de un mapa de entorno de entrada, y de muestrear direcciones de luz de forma eficiente; y una red neuronal implícita con activaciones sinusoidales, capaz de transformar direcciones de luz a valores de radiancia lineal en RGB. Demostramos la eficacia de nuestro método en aplicaciones de muestreo por importancia para renderización por

Monte Carlo, consiguiendo representaciones de iluminación más precisas y exactas que métodos anteriores de compresión de iluminación; y dos órdenes de magnitud menor coste computacional que métodos tradicionales de muestreo de iluminación. (Capítulo 7).

- Un estudio integral de las capacidades y limitaciones de las redes neuronales profundas para el problema de la descomposición intrínseca. Mediante un estudio exhaustivo del estado del arte, los datasets, funciones de coste, arquitecturas neuronales y heurísticas, proporcionamos una nueva categorización de estos trabajos. Además, proponemos nuevas direcciones de investigación, construyendo sobre artículos recientes en renderización neuronal, inverso o diferenciable; además de los nuevos avances en sistemas de aprendizaje automático. Esta contribución está publicada en [Gar+22], y usamos parte de ella para la sección 1.1.

Estas contribuciones han dado lugar a las siguientes publicaciones:

- *"Neural Photometry-guided Visual Attribute Transfer"*
Carlos Rodríguez-Pardo, Elena Garces; *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2021 [RG21].
Este journal tiene un factor de impacto de 5.226, su posición en el índice JCR es 13 de 110 (Q1) en la categoría Computer Science, Software Engineering (datos de 2021).
- *"A Survey on Intrinsic Images: Delving Deep into Lambert and Beyond"*
Elena Garces, Carlos Rodríguez-Pardo, Dan Casas, Jorge Lopez-Moreno; *International Journal in Computer Vision (IJCV)*, 2022 [Gar+22].
Este journal tiene un factor de impacto de 13.369, su posición en el índice JCR es 10 de 145 (Q1) en la categoría Computer Science, Artificial Intelligence (datos de 2021).
- *"SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps"*
Carlos Rodríguez-Pardo, Elena Garces; *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2022 [RG22].
Este journal tiene un factor de impacto de 5.226, su posición en el índice JCR es 13 de 110 (Q1) en la categoría Computer Science, Software Engineering (data from 2021)
- *"How Will It Drape Like? Capturing Fabric Mechanics From Depth Images"*
Carlos Rodríguez-Pardo, Melania Prieto-Martin, Dan Casas, Elena Garces; *Computer Graphics Forum (Proceedings of Eurographics 2023)* [Rod+23b].
Este journal tiene un factor de impacto de 2.363, su posición en el índice JCR es 53 de 110 (Q2) en la categoría Computer Science, Software Engineering (datos de 2021).

- *"UMat: Uncertainty-Aware Single Image High Resolution Material Capture"*
Carlos Rodríguez-Pardo, Henar Domínguez, David Pascual, Elena Garces;
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023 [Rod+23a].
Esta conferencia es el evento de mayor impacto en ciencias de la computación según *Google Scholar*, y está categorizada como CORE A++ conference (datos de 2021)).
- *"Towards Material Digitization with a Dual-scale Optical System"*
Elena Garces, Víctor Arellano, Carlos Rodríguez-Pardo, David Pascual, Sergio Suja, Jorge López-Moreno; *ACM Transactions on Graphics (Proceedings of SIGGRAPH), 2023* [Gar+23].
Este journal tiene un factor de impacto de 7.403, su posición en el índice JCR es 9 de 110 (Q1) en la categoría Computer Science, Software Engineering (datos de 2021).
- *"NeuBTF: Neural Fields for BTF Encoding and Transfer"*
Carlos Rodríguez-Pardo, Konstantinos Kazatzis, Jorge López-Moreno, Elena Garces; 2023
Esta publicación se encuentra actualmente en revisión.
- *"NEnv: Neural Environment Maps for Global Illumination"*
Carlos Rodríguez-Pardo, Javier Fabre, Elena Garces, Jorge López-Moreno; 2023
Esta publicación se encuentra actualmente en revisión.

A.5 Conclusiones

A lo largo de esta tesis, se han abordado diferentes retos en los campos de la informática gráfica, el aprendizaje automático y la visión artificial; desde la generación de representaciones neuronales más eficientes para la codificación de componentes de las escenas, hasta el diseño de algoritmos de aprendizaje profundo para la digitalización de materiales a bajo coste y alta precisión. En concreto, se han propuesto algoritmos para la propagación de atributos visuales (Capítulo 2), para la digitalización realista de la reflectancia de materiales (Capítulo 2.C), su edición y síntesis hacia texturas teselables (Capítulo 3), o su codificación mediante campos neuronales (Capítulo 4). Además, se han propuesto sistemas de captura escalables y de alta calidad mediante dispositivos de bajo coste, incluyendo escáneres para la captura de reflectancia (Capítulo 5) o las cámaras de profundidad para la captura de parámetros mecánicos de tejidos (Capítulo 6). Finalmente, se ha propuesto una representación neuronal capaz de introducir iluminación global a escenas virtuales de forma diferenciable y computacionalmente más eficiente y precisa

que las soluciones disponibles en la literatura (Capítulo 7). Los resultados de esta tesis forman parte de ocho publicaciones, seis de las cuales están aceptadas en revistas académicas o conferencias internacionales, mientras que las demás se encuentran en proceso de revisión.

En conjunto, en esta tesis se introducen avances significativos para la representación y generación de escenas virtuales realistas, proporcionando métodos basados en aprendizaje automático capaces de mejorar el realismo y la eficiencia de estas escenas. El impresionante ritmo de la investigación en inteligencia artificial dificulta prever las soluciones que se desarrollarán en el futuro, sin embargo, creemos que la combinación de soluciones basadas en datos con la renderización basada en la física seguirá definiendo el campo en los años venideros. Esperamos que las ideas presentadas en esta tesis contribuyan a la generación de soluciones digitales más accesibles y sostenibles medioambiental y socialmente.



Tesis doctoral

Carlos Miguel Rodríguez-Pardo Jiménez

Tesis doctoral