Workshop on Material Appearance Modeling (2019)
H. Rushmeier and R. Klein (Editors)

Neural Appearance Synthesis and Transfer

Ilya Mazlov[1], Sebastian Merzbach[1], Elena Trunz[1] and Reinhard Klein[1]

[1]University of Bonn, Germany

Abstract

Appearance acquisition is a challenging problem. Existing approaches require expensive hardware and acquisition times are long. Alternative "in-the-wild" few-shot approaches provide a limited reconstruction quality. Furthermore, there is a fundamental tradeoff between spatial resolution and the physical sample dimensions that can be captured in one measurement. In this paper, we investigate how neural texture synthesis and neural style transfer approaches can be applied to generate new materials with high spatial resolution from high quality SVBRDF measurements. We perform our experiments on a new database of measured SVBRDFs.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Capturing the appearance of real surfaces requires scanning in the spatial and the bi-angular domain of light and view directions. As the reflectance of most materials shows high-dynamic-range properties, this is an involved process that requires carefully calibrated cameras and light sources. There are commercial devices available [XR18], but at a high cost. Recent trends show an application of deep learning for tackling the severely ill-posed problem of few-shot reflectance acquisition [YLD*18, DAD*18, LXR*18]. However, these approaches are limited in model complexity and general reconstruction quality. Merzbach et al. [MHRK19] predict high quality complex SVBRDF parameters, but they still require dense, calibrated inputs. For our approach we rely on an existing corpus of high quality SVBRDFs, e.g. the publicly available fabric samples in the Bonn Fabric SVBRDF dataset[†] [MHRK19]. We adapt two deep-learning-based methods, the texture synthesis method of Zhou et al. [ZZB*18] and the Neural Style Transfer by Gatys et al. [GEB15b] to SVBRDF materials from this database. Our work has the following contributions

- example-based synthesis of higher-resolution SVBRDFs of material samples with limited spatial resolution;
- appearance transfer of existing to new target materials;
- re-use of existing RGB-pre-trained CNN features without the need for costly re-training on materials.

† https://cg.cs.uni-bonn.de/svbrdfs/

2. Related Work

Neural style transfer and neural texture synthesis are the two branches of works underlying our paper. Example-based texture synthesis deals with the problem of creating spatially enlarged instances of small exemplars of a texture. During style transfer the artistic style of an input image is transferred to the semantic structures of a content image by optimizing a style loss. Texture synthesis and style transfer are closely related. An extensive overview of existing neural style transfer and neural texture synthesis approaches is provided by Jing et al. [JYF*19]. Gatys et al. were the first who proposed a deep learning approach for texture modelling [GEB15a] and extended their ideas in a subsequent work to the transfer of style of paintings to other "content" images [GEB15b]. These approaches work by passing a style image through a pre-trained convolutional neural network (CNN) and computing Gramian matrices on the features of some of the convolutional layers. To produce a new instance of a style image applied to an additional provided content image, an optimization is run over the output image, which is initialized with noise. The optimization tries to progressively minimize the style loss that enforces similar Gram matrices between style and output images, and a content loss that enforces similar features on another subset of the CNN layers between output and content image. In a more recent work, Zhou et al. [ZZB*18] achieve state of the art texture synthesis results using a generative adversarial network (GAN) in combination with a style loss.

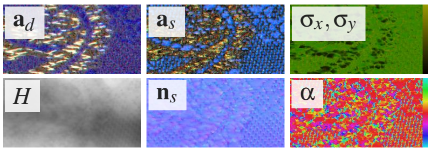

Figure 1: *Example material maps from the Bonn Fabric SVBRDF dataset. Base and highlight colors are respectively defined by diffuse ($\mathbf{a}_d$) and specular ($\mathbf{a}_s$) albedos, the glossiness by the roughness parameters σ_x, σ_y (displayed in R and G channels), displacement (H) and shading normal $\mathbf{n}_s$ encode fine-scale surface variations, and the anisotropy angle α (color-coded) defines the dominant anisotropy direction.*

All of these methods work exclusively on RGB images. This is because they re-use CNN models pre-trained on large-scale image datasets. Since our material representations contain more then 3 channels, we cannot simply feed them as input to the existing models. Naïve splitting into 3-channel images which are fed individually will produce uncorrelated results that cannot simply be concatenated. We therefore need to adapt these models to our special multi-channel inputs, ideally without having to train the underlying CNNs from scratch. The latter would pose very challenging because of a lack of training data.

Material Model: We briefly describe the SVBRDF inputs that we are processing. The database we use for our experiments contains fabric samples represented using the Geisler-Moroder variant [GMD10] of the anistropic Ward BRDF [W*92], extended by a Fresnel term based on the Schlick approximation [Sch94]. For a detailed description of model the reader is referred to the original works or Merzbach et al. [MHRK19]. The model parameters, represented in individual texture maps to allow spatial variations across the surface, are shown in Fig. 1.

3. Neural SVBRDF Synthesis

Zhou et al. [ZZB*18] introduce an example-based texture synthesis that – contrary to many previous works – allows to generate textures with non-stationary characteristics. Their results are very appealing and motivate the application to the fabric SVBRDFs in our database, many of which show exactly these properties. As it is designed for RGB textures only, we have to adapt the method in the following ways: We change the network architecture to allow for more than the 3 RGB channels as inputs. This change is straightforward except for the computation of the style loss. The underlying VGG network [SZ14] is pretrained on RGB images only and cannot simply be replaced by an equivalent architecture with more input channels. We solve this problem by splitting the SVBRDFs into m 3-channel textures (see below), which we can directly pass through VGG-net. The resulting feature maps are then concatenated along the feature-dimension. Finally, the Gram matrices can be calculated in the same principle as before, only that ours are m times bigger. Accordingly, we have to adjust the normalization weight for the style loss to account for the additional factor of m^2.

We apply the following mappings to our input to facilitate learning: The lobe parameters σ_x, σ_y are highly non-linear, so we transform them via $\sigma' = \frac{\log(\sigma+0.001)-\log(0.001)}{\log(0.65)-\log(0.001)}$. The anisotropy angle $\alpha \in [-\pi/2, \pi/2]$ shows discontinuities when it wraps around, which causes high contrast in the parameter map, when in reality the observed effect on the reflectance is only very subtle. We therefore transform α to a 2D representation $\alpha \mapsto \{\sin(2\alpha), \cos(2\alpha)\}$. After these transformations the SVBRDFs are represented with 14 channel textures. We furthermore increase the training efficiency by normalizing the different modalities in the parameter maps. We empirically found that a channel-wise normalization with the 0.1-th and 99.9-th percentiles provides the best results.

We split the 14 channels of the mapped parameters into $m = 8$ separate RGB images by grouping semantically related parameters, repeating some of them to obtain 3-channel textures ($\mathbf{a}_d$, $\mathbf{a}_s$, $\mathbf{n}_d$, $3\times\sigma_x$, $3\times\sigma_y$, $3\times\sin(2\alpha)$, $3\times\cos(2\alpha)$, $3\times H$). We also experimented with $m = 6$ maps by respectively concatenating the lobe and anisotropy parameters but obtained slightly better results with the above version.

4. Neural Appearance Transfer

Our adaption of the texture synthesis method to SVBRDFs provides good results. However, in most cases it is desirable to have more control over the synthesized materials. Inspired by the texture *transfer* experiments presented by Zhou et al. we also investigate neural style transfer methods on SVBRDFs. Zhou's texture transfer experiments provide promising results. However, we found it difficult reproducing similar results with our adapted implementation on materials. Furthermore, the method has a significant training overhead of several hours for each material.

So instead we focused on image optimization based neural style transfer methods. These methods achieve, in comparison to model optimization based methods, more appealing results [JYF*19]. Furthermore, they require much lower training effort, as there is no GAN component as in Zhou's network that drives up the training costs. We therefore select the neural style transfer method by Gatys et al. [GEB15b] because of its simplicity and adapt it to allow for appearance transfer. When trying to extend it to our 14 channel SVBRDF representation, we face the same problem as with the texture synthesis method of Zhou et al. [ZZB*18]. We can thus apply the same strategy of grouping semantically related texture maps into 3-channel images, which we individually pass through the VGG-net, and concatenate the resulting feature maps to compute the Gram matrices. Similar changes allow computing the content loss term on the entire content-SVBRDF.

Not all of the SVBRDF parameters are equally "important" for the resulting appearance. The variations stored in the displacement map have a much less noticeable impact on a rendering than e.g. the albedo maps. Similarly, even with our parameter mappings and layer-wise normalization, some features are less prominent and cause different degrees of feature activations in the CNNs. Some parameters are much less correlated with the others, most noticeably the displacements H. We therefore introduce a weighting scheme into our style loss calculation, which applies different weights to the different parameter types. It emphasizes the albedo maps and decreases the weight for the displacement map using a weight vector $w = [1.5, 2, 1, 1, 1, 0.05]$. This vector also requires normalization in order not to shift the style-content loss balance. The normalization is given by ${}^{m}/_{||w||_1}$.

5. Results

Appearance synthesis: In the following we first present results for appearance synthesis based on our adaption of Zhou et al. [ZZB*18] on a set of various materials, see Figs. 2 and 3. We generally obtain visually appealing results after around 50000 training iterations.

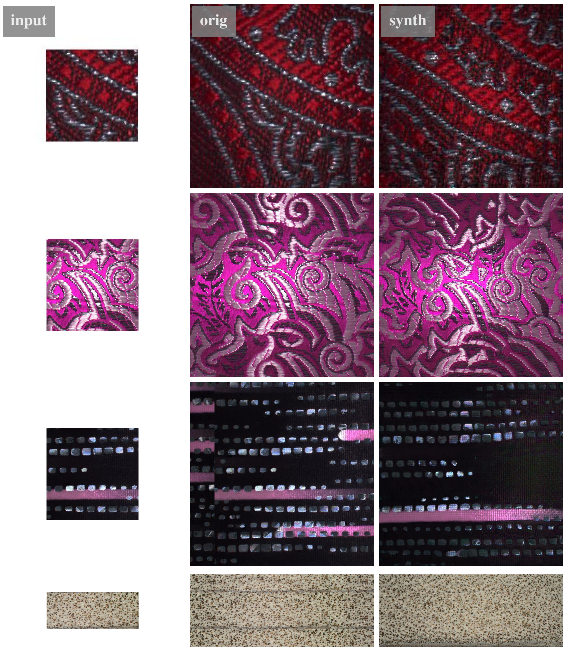

Figure 2: *Result renderings of our texture synthesis. Columns from left to right show:* ***input:*** *low resolution crop,* ***orig:*** *original uncropped material with input patch in the center,* ***synth:*** *synthesized high resolution material.*

Appearance transfer: Next, we show transfered appearance based on our adapted neural style transfer method [GEB15b]. Figs. 4 shows renderings of style SVBRDFs transfered according to content SVBRDFs.

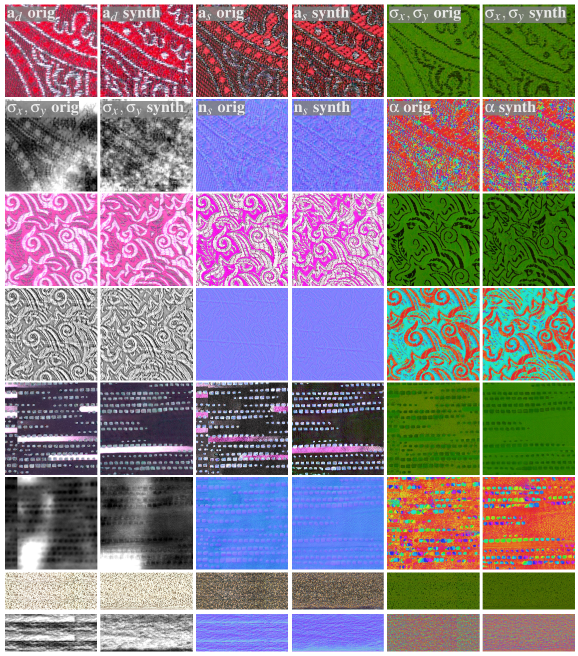

Figure 3: *Parameter maps of original and synthesized materials. These results correspond to the renderings in Fig. 2.*

6. Conclusion

In this paper we first apply the ideas of the approach of Zhou et al. [ZZB*18] to the problem of synthesis of high resolutions SVBRDFs of material samples. The synthesized textures with increased resolution look appealing and perceptually similar (including all reflectance properties) to the original materials, while preserving global structures. However, the resolution can only be extended by the fixed factor between low and high resolution training samples. Second, we extend the approach of Gatys et al. [GEB15b] to the task of appearance transfer of fabrics. Though our implementation is not yet universally applicable to all combinations of style and content materials, we still achieve very promising results.

We plan to further investigate improved weighting schemes to stabilize the behavior in a future work. Furthermore, the next obvious extension is to relax the need for a content *SVBRDF* and the approach to handle arbitrary RGB content images.

References

[DAD*18] Deschaintre V., Aittala M., Durand F., Drettakis G., Bousseau A.: Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 128. 1

[GEB15a] Gatys L., Ecker A. S., Bethge M.: Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems* (2015), pp. 262–270. 1

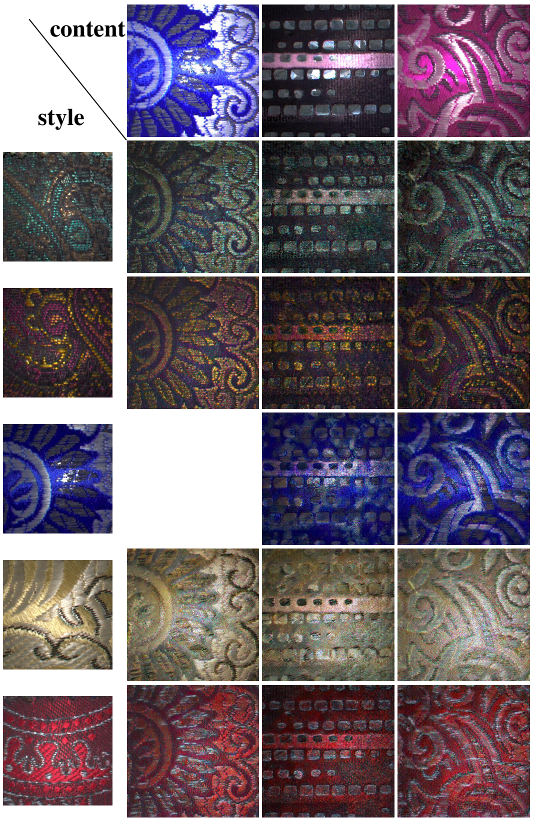

Figure 4: *Result renderings of our neural appearance transfer. The left column shows style SVBRDFs transfered respectively according to content materials shown in the top row.*

[GEB15b] GATYS L. A., ECKER A. S., BETHGE M.: A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015). 1, 2, 3

[GMD10] GEISLER-MORODER D., DÜR A.: A new ward brdf model with bounded albedo. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 1391–1398. 2

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in neural information processing systems* (2014), pp. 2672–2680. 4

[JYF*19] JING Y., YANG Y., FENG Z., YE J., YU Y., SONG M.: Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics* (2019). 1, 2

[LXR*18] LI Z., XU Z., RAMAMOORTHI R., SUNKAVALLI K., CHANDRAKER M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia 2018 Technical Papers* (2018), ACM, p. 269. 1

[MHRK19] MERZBACH S., HERMANN M., RUMP M., KLEIN R.: Learned fitting of spatially varying brdfs. In *Proceedings of the Eurographics Symposium on Rendering* (2019), Eurographics Association. 1, 2

[Sch94] SCHLICK C.: An inexpensive brdf model for physically-based rendering. In *Computer graphics forum* (1994), vol. 13, Wiley Online Library, pp. 233–246. 2

[SCO17] SENDIK O., COHEN-OR D.: Deep correlations for texture synthesis. *ACM Transactions on Graphics (TOG) 36*, 5 (2017), 161. 4

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014). 2, 4

[W*92] WARD G. J., ET AL.: Measuring and modeling anisotropic reflection. *Computer Graphics 26*, 2 (1992), 265–272. 2

[XR18] X-RITE: Tac7-scanner, June 2018. URL: `http://web.archive.org/web/20180615015942/https://www.xrite.com/categories/appearance/tac7`. 1

[YLD*18] YE W., LI X., DONG Y., PEERS P., TONG X.: Single image surface appearance modeling with self-augmented cnns and inexact supervision. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 201–211. 1

[ZZB*18] ZHOU Y., ZHU Z., BAI X., LISCHINSKI D., COHEN-OR D., HUANG H.: Non-stationary texture synthesis by adversarial expansion. *arXiv preprint arXiv:1805.04487* (2018). 1, 2, 3, 4

7. Appendix

Addition of Deep Correlation loss:

In another recent work, Sendik and Cohen-Or introduced a Deep Correlation loss [SCO17] for their neural texture synthesis method. It enables a better synthesis of textures that show regular structures. Given that this property applies to many fabrics, we investigate the impact of adding this loss term to the synthesis method of Zhou et al. [ZZB*18]. We thus augment their total loss function presented by an additional term $\mathcal{L}_{DCorr}$ that is computed according to the Deep Correlation loss [SCO17]:

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \lambda_1 \mathcal{L}_{L_1} + \lambda_2 \mathcal{L}_{style} + \lambda_3 \mathcal{L}_{DCorr}, \quad (1)$$

where $\mathcal{L}_{adv}$ is an adversarial loss [GPAM*14], $\mathcal{L}_{L_1}$ a simple L_1 loss, and $\mathcal{L}_{style}$ a style loss computed on VGG-19 [SZ14].

Fig. 5 shows the effects of the addition of the Deep Correlation loss term, as well the difference when it completely replaces the style loss. Both when augmenting and replacing the style loss, we observe qualitatively comparable results. Since the calcuation of the correlation matrices is quite costly, the training performance drops by a factor of 5. We conclude that Deep Correlation loss poses an interesting alternative to the style loss, however, the performance penalties outweigh the potential benefits.

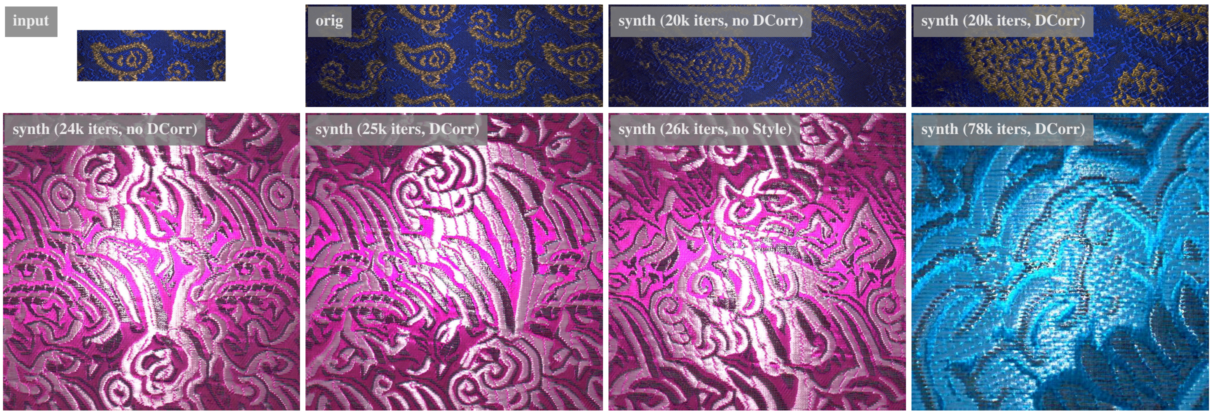

Figure 5: *Ablation study for the effect of an additional Deep Correlations term (see Eq. 1).* ***Top row:*** *input, uncropped material, synthesis with style loss only, synthesis with style and deep correlation loss;* ***bottom row (pink fabric):*** *further synthesis results with style loss only, with style and deep correlation loss, and with deep correlation loss only;* ***bottom right (green-blue fabric):*** *typical artifacts observed when using deep correlation loss.*