

Controlling Material Appearance by Examples

Yiwei Hu^{1,2}, Miloš Hašan¹, Paul Guerrero¹, Holly Rushmeier² and Valentin Deschaintre¹

¹ Adobe Research

² Yale University

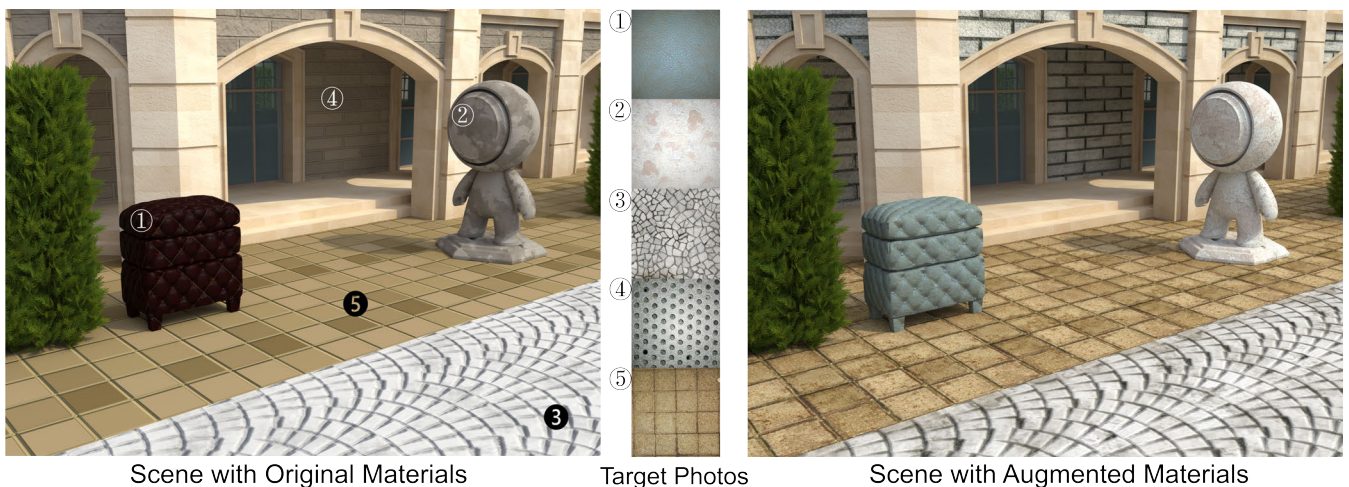


Figure 1: We propose a method to control and improve the appearance of an existing material (left) by transferring the appearance of materials in one or multiple target photo(s) (center) to the existing material. The augmented material (right) combines the coarse structure from the original material with the fine-scale appearance of the target(s) and preserve the input tileability. Our method can also transfer appearance from materials from different types by spatial control. This enables a simple workflow to make existing materials more realistic using readily-available images or photos.

Abstract

Despite the ubiquitous use of materials maps in modern rendering pipelines, their editing and control remains a challenge. In this paper, we present an example-based material control method to augment input material maps based on user-provided material photos. We train a tileable version of MaterialGAN and leverage its material prior to guide the appearance transfer, optimizing its latent space using differentiable rendering. Our method transfers the micro and meso-structure textures of user provided target(s) photographs, while preserving the structure and quality of the input material. We show our methods can control existing material maps, increasing realism or generating new, visually appealing materials.

CCS Concepts

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

1. Introduction

Realistic materials are one of the key components of vivid virtual environments. Materials are typically represented by a parametric surface reflectance model and a set of 2D texture maps (material maps) where each pixel represents a spatially varying parameter of the model (e.g. albedo, normal, and roughness values). This representation is ubiquitous because of its compactness and ease of vi-

sualization. It is also used by most recent light-weight acquisition methods [DAD*18, DAD*19, GLD*19, GSH*20]. Material maps are however difficult to edit. In addition to the significant artistic expertise required to create realistic material detail, tools such as Photoshop [Ado22a] are designed for natural images rather than parameter maps and make it hard to account for the correlations that exist between individual material maps. Previous work pro-

posed propagating local edits globally through segmentation and similarity [AP08] or providing an SVBRDF[†] exemplar to transfer material properties [DDB20], but these edits remain limited to spatially constant material parameter modifications. Procedural materials [Ado22b, HDR19, SLH*20, HHD*22] also enable material map editing, but are challenging to design and can lack realism if the procedural model is not expressive and realistic enough [HDR19].

In this paper, we propose a method to control the appearance of material maps using photographs or online textures. Given a set of input material maps and target photographs (or textures), we transfer micro- and meso-scale texture details from the target photographs to the input material maps. We preserve the large-scale structures of the input material, but augment the fine-scale material appearance based on the target photographs using a variant of style transfer [GEB16]. Unlike Deschaintre et al. [DDB20], our approach uses easy to find or capture photographs instead of SVBRDF maps as targets.

Using style transfer from photographs to material maps poses several challenges. First, we need to optimize the material maps to produce the desired fine-scale details in the rendered image, rather than directly optimizing the output image. Second, we need to make sure material properties remain realistic during this optimization, including the correlation between material maps. In Fig. 2, we show that a naive adaptation of image style transfer fails to generate high-quality material maps due to these difficulties. To address both of these challenges, we propose using material generation priors to guide the transfer of micro and meso-scale texture details while retaining realistic material properties. We use MaterialGAN, a generative model trained on material maps [GSH*20], as a prior and modify it to preserve *tileability*, a particularly important feature to keep memory requirement low in production. Our transfer operation optimizes the latent space of our pre-trained tileable MaterialGAN using differentiable rendering to transfer the relevant details while preserving the structure of the original input material maps.

The idea of material appearance or style on a spatially-varying material can be ambiguous. There are different aspects of appearance that a designer may see in an exemplar that they would like to transfer to an existing material, requiring some control. To provide this control we introduce multi-example transfer with spatial guidance: our method allows transferring texture details to precise user-specified regions only. We propose using a sliced Wasserstein loss [HVCB21] to guide our transfer and supports multi-target transfer thanks to a resampling strategy we introduce. We additionally describe a slightly slower but better grounded formulation to compare distributions with different sampling based on the Cramér loss [Cra28].

We demonstrate creating new materials with our image-guided editing operations in various applications. To summarize, our contributions are:

- A material transfer method for controlling appearance of material maps using photo(s).

- MaterialGAN as a prior for tileable materials appearance transfer.
- A multi-target transfer option with fine region control.

2. Related Work

2.1. Material Appearance Control

Control and editing of material appearance is a long standing challenge in computer graphics. Different solutions have been proposed, depending on the targeted material representation. Tabulated materials are represented in a very high dimension, unintuitive space [WAKB09], making their manipulation difficult. Lawrence et al. [LBAD*06] represent a spatially-varying (SV) measured materials as an inverse shade tree, decomposing it into spatial structure and basis BRDFs to facilitate editing through extracted "1D curves" representing physical directions. Ben-Artzi et al. [BAOR06, BAEDR08] proposed a fast iteration editing framework of material "in situ", allowing to efficiently visualize global illumination effects. Lepage et al [LL11] proposed material matting to decompose measured SV-BRDF into layers for spatial editing. More recently, Serrano et al. [SGM*16] and Shi et al [SWSR21] proposed designing perceptual spaces for more intuitive BRDF editing and Hu et al. [HGC*20] proposed encoding the BRDF in a deep network, reducing its dimensionality and simplifying editing. These methods target measured BRDF editing and focus on extracting relevant dimensions (perceptual, spatial) along which globally uniform edits can be made. As opposed to these methods, our approach targets spatially varying analytical model editing, enabling complex spatial detail transfer.

Analytical BRDFs represent materials based on pre-determined models (e.g Cook-Torrance [CT82], Phong [Pho75], GGX [WMLT07a], etc.) and their parameters. To explore this parameter space, Ngan et al. [NDM06] propose an interface to navigate different BRDF properties with perceptually uniform steps and Talton et al. [TGY*09] leverage a collaborative space to define a good modeling space for users to explore. Image-space editing was also explored with gloss editing in lightfields [GMD*16] and material properties modification [ZFWW20]. Recently, different methods were proposed to optimize or create procedural materials [HDR19, SLH*20, HHD*22]. While procedural materials are inherently editable, they are limited to the expressivity of their node graphs and are difficult to design. More closely related to our approach are image-guided material properties and style transfer. Fiser et al. [FJL*16] used drawing on a known shape (sphere) to transfer the style and texture to a more complex drawing. Deschaintre et al. [DDB20] proposed using a surface picture and material exemplars to create a material with the surface picture structure and exemplar properties. Recently, Rodriguez-Pardo et al. [RPG22] proposed leveraging photometric inputs to transfer material properties annotated on a small region to larger samples. As opposed to these methods our approach transfers the appearance and texture micro and meso structures from photo(s) to a pre-existing analytical SVBRDF.

2.2. Style Transfer

We formulate our by-example control on material maps as a material transfer problem. In recent years, neural style transfer [GEB16]

[†] Spatially-Varying Bidirectional Reflectance Distribution Function

and neural texture synthesis [GEB15] have been used in a variety of contexts (e.g. sketching [TTK*21, SJT*19], video [JvST*19, TFK*20], painting style [TFF*20]). These methods are based on the matching of the statistics extracted by a pre-trained neural network between output and target images. For example, Gatys et al. [GEB15, GEB16] leverage a pre-trained VGG neural network [SZ15] to guide style transfer, using the Gram Matrix of extracted deep features from the image as their statistical representation. Heitz et al. [HVCB21] described an alternative sliced Wasserstein loss as a more complete statistical description of extracted features. Different approaches proposed to train a neural network to transfer style of images or synthesize textures in a single forward operation [JAFF16, ULVL16, HB17, ZZB*18]. These methods however focus on the transfer of the overall style of images and do not account for details. Domain specific (faces) and guided style transfer have been proposed [KSS19, HZL19] to better control the process and transfer style between semantically compatible parts of the images.

In the context of materials, Nguyen et al. [NRM*12] proposed transferring the style or mood of an image to a 3D scene. They pose this problem as a combinatorial optimization of assigning discrete materials, extracted from the source image, to individual objects in the target 3D scene. Mazlov et al. [MMTK19] proposed directly applying neural style transfer on material maps in a cloth dataset using a variant of RGB neural style transfer [GEB16].

In contrast to this previous work, we focus on transferring micro- and meso-scale details from photos onto 2D material maps and enable transfer spatial control. In particular, we differ from Mazlov et al. [MMTK19] in our inputs. Our style examples are simple photographs, easily captured or found online. Our approach allows us to deal with the ambiguity of the material properties in the target photo through material priors and differentiable rendering while preserving the input material map structure and eventual tileability.

2.3. Procedural Material Modeling

As mentioned, procedural materials are inherently editable as they rely on parametrically controllable procedures. Recent work on inverse procedural material modeling [HDR19, GHYZ20, SLH*20, HHD*22] aim at reproducing material photographs, but are still limited to simple or existing procedural materials. While editable, these models remain challenging to create and their optimization relies on the expressiveness of the available procedural materials, with no guarantee that a realistic model exists [HDR19, SLH*20]. Often procedural materials can look synthetic because they are not complex enough. In such cases, our method can augment the material appearance by transferring detailed real materials onto the synthetically generated material maps, while preserving their tileability. In this context, Hu et al. [HDR19] proposed to improve the quality of fitted materials through a style augmentation step, but style was only transferred to the final rendered images and did not impact the material maps themselves.

2.4. Material Acquisition

As an alternative to material transfer, one could directly capture the target material. Classical material acquisition methods require dozens to thousands of material samples to be captured under controlled illuminations [GGG*16]. Aittala et al. [AAL16] proposed using a single flash image of a stationary material to reconstruct a patch of it through neural guided optimization. Recently, deep learning was used to improve single [LDPT17, DAD*18, HDMR21, GLT*21, ZK21] and few-images [DAD*19, GSH*20, GLD*19, YDPG21] material acquisition. These methods recover 2D material maps based on an analytical BRDF model e.g. GGX [WMLT07b]. As they mainly focus on acquisition, they do not enable control. As opposed to these methods, our approach provides a convenient way to control the appearance of material maps, leveraging the input material to retain high quality, allowing for multi-target control and preserving important properties such as tileability. Additionally, our method does not require a flash photograph as target, allowing the use of internet-searched references.

3. Method

Our method transfers fine-scale details from user-provided target material photos onto a set of material maps. The material maps contain diffuse albedo, normal, roughness and specular albedo properties, encoding parameters for a GGX [WMLT07b] shading model. Since fine-scale details are typically underdefined by single-view photos, we choose to leverage a specialized prior that regularizes the transfer of materials details to ensure our material maps remain realistic and to help disambiguate unclear material properties in the target photo.

As prior, we train a tileable version of MaterialGAN [GSH*20] using a large dataset of synthetic material maps. This prior is described in Section 3.1. We transfer material appearance by optimizing our material maps in the latent space of this tileable MaterialGAN, rather than directly in pixel space. A differentiable GGX shading model renders the optimized material maps into an image where they are compared to the target photos. This appearance transfer approach is described in Section 3.2. In Section 3.3, we describe how we add spatial control over the appearance transfer and enable transfer from multiple targets using label maps and a resampled version of the sliced Wasserstein loss [HVCB21].

3.1. Tileable MaterialGAN

MaterialGAN [GSH*20] has been shown to be a good prior for lightweight material acquisition. However, the original MaterialGAN, based on StyleGAN2 [KLA*20], is not designed to produce tileable outputs. It has slight artifacts on the borders, even if the training data is perfectly tileable. Moreover, material maps in most of publicly available material datasets [DAD*18, DDB20] are not tileable. To address this limitation, we modify StyleGAN's architecture following recent insights in GAN designs [KAL*21, ZHD*22] to ensure tileability of the synthesized material maps, even if the training data is not tileable. Specifically, we prevent the network from processing the image borders differently from the rest of the image, by modifying all convolutional and upsampling operations with circular padding.

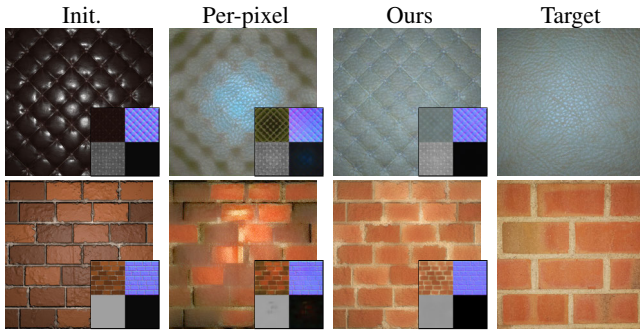


Figure 2: Comparison between optimizing the material maps in pixel space and our approach, optimizing in the latent space of tileable MaterialGAN. Insets show the material maps used to render each image and Target is a photograph representing the desired appearance. Per-pixel optimization frequently gets trapped in local minima and suffers from artifacts and light-baking, while our use of a prior allows to reach the desired appearance.

Once we enforce the generator network to always produce tileable outputs, we cannot show tileable synthesized and non-tileable real data to the discriminator, as it would have a clear signal to differentiate them. Instead, as suggested by [ZHD*22], we randomly crop both real and synthesized material maps. The discriminator cannot identify whether the crop comes from a tileable source or not, and instead has to identify whether the crop content looks like a real or fake material.

We train our model with the same loss functions as StyleGAN2 [KLA*20] including cross-entropy loss with R1 regularization and path regularization loss for generators. See Sec. 4 for training details. With the trained tileable material prior, our material transfer method overcomes the local minimum problems caused by simple per-pixel optimization (Fig. 2) and reconstructs high-quality material maps compared to an adaption of Deep Image Prior (Fig. 12). In Fig. 3, we show our modified network architecture can successfully preserve tileability after transfer compared to the original MaterialGAN. Importantly, the preserved tileability allows us to directly apply our transfer materials on different objects seamlessly in a large-scale scene as shown in Fig. 1.

3.2. Neural Material Transfer

In previous work, Hu et al. [HDR19] proposed an extra style augmentation step to add realistic details on their fitted procedural materials given a photo as target. However, their transfer is only applied to the rendered images and does not modify the material maps, limiting the impact of their transfer step (e.g. the material can not be relit). Different from their method, we directly transfer the target photo(s) material appearance onto material maps, allowing the use of the new material in a traditional rendering pipeline.

Given a set of input material maps M_0 and a user-provided target image I , we compute the transferred material maps M as follows:

$$\arg \min_M d_0(R(M), I) + d_1(M, M_0) \quad (1)$$

where R is a differentiable rendering operator, rendering material maps M into an image. $d_0(R(M), I)$ measures the statistical

similarity between the synthetic image $R(M)$ and target image I . $d_1(M, M_0)$ is a regularization term that penalizes the structure difference between transferred material maps M and the original input M_0 . The lighting used in R can easily be adapted if needed to roughly match the lighting conditions of target images. However, we found that a simple co-located point light works well in the examples we tested. Similar to neural style transfer and neural texture synthesis, we apply a statistics-based method to measure the similarity for d_0 and d_1 . Common choices for d_0 and d_1 are style loss and feature loss [GEB15]. However, simply performing per-pixel optimization on material maps M (i) fails to reach the appearance of the target photograph. This is caused by challenging local minima in the optimization and a high sensitivity to the learning rate, requiring careful tuning (see Figure 2). And (ii) the optimization may result in departure from the manifold of realistic materials. Instead, we take advantage of the learned latent space of our pre-trained tileable MaterialGAN to regularize our material transfer, effectively solving these problems.

We tackle the material transfer challenges in two steps. First, we project the input material maps M_0 into the latent space of the pre-trained MaterialGAN model f by optimizing the latent code θ . The optimization is guided by L_1 loss and feature loss:

$$L_\theta = \|f(\theta) - M_0\|_1 + \|F(f(\theta)) - F(M_0)\|_1 \quad (2)$$

where F is a feature extractor using a pre-trained VGG network [SZ15]. With the projected latent vector, we perform material transfer by optimizing θ to minimize the statistical difference between rendered material $R(f(\theta))$ and the material target I

$$L_\theta = \|S(R(f(\theta))) - S(I)\|_1 + \|F(f(\theta)) - F(M_0)\|_1 \quad (3)$$

The statistical descriptor S , the style loss, can be implemented in different ways. The Gram Matrix [GEB15] is a commonly used statistical loss. Recently, Heitz et al. [HVCB21] proposed a sliced Wasserstein loss to measure a more complete statistical difference, taking additional statistical moments into consideration. Their implementation however relies on comparing signals (here images) with the same number of samples (here pixels). To remove this limitation, we propose using uniform resampling to balance the number of samples in both distributions to facilitate spatial control and multi-target transfer (Sec. 3.3). We derive in supplemental material a slightly slower but better grounded sliced Cramér loss, comparing the distribution CDFs rather than PDFs.

3.3. Spatial Control and Multi-target Transfer

The sliced Wasserstein loss has been shown to be a good statistical metric [HVCB21] to compare deep feature maps, but does not allow comparing feature maps with different numbers of samples (pixels) trivially. The "tag" trick introduced in the original paper cannot be applied to our goal for multi-target transfer. The sliced Wasserstein loss compares two images by projecting per-pixel VGG feature vectors onto randomly sampled directions in feature space, giving two sets of 1D sample points u and v , one for each image. These are compared by taking the difference between the sorted sample points. To allow for different sample counts



Figure 3: The original MaterialGAN model does not produce tileable results, leading to visible seams on the boundary when tiled, while our tileable network preserves the input tileability without artifacts (Ours). Seams are particularly visible when zoomed in.

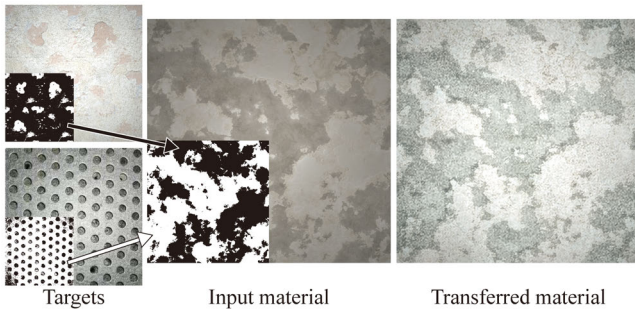


Figure 4: For our multi-target transfer, we transfer the appearance of two targets to the same input material. A binary map on the input material defines which of the two targets should be used for each pixel. Additionally, a binary map can be defined on each target defines which regions should be included in the appearance transfer. Here we transfer the appearance from the black region of the mask of the top target to the black regions on the input mask. The white regions are similarly transferred from the bottom target. This binary system is only used for visualization. There is no limit to the number of correspondences that can be made.

$|u| < |v|$, we introduce the *resampled* sliced Wasserstein loss as:

$$L_{SW1D}(u, v) = \frac{1}{|u|} \|\text{sort}(u) - \text{sort}(U(v))\|_1 \quad (4)$$

where $U(v)$ is an operator that uniformly random subsamples a vector to obtain $|u|$ samples from v . Note here we compute L_1 error as opposed to squared error suggested in the original paper as we found it to perform better.

Using this resampling approach, we can compute statistical differences between labeled regions of different size. Assume we have label maps associated with material maps and each target photo, we define our transfer rule as **Label X: Target Y, Z** which means to transfer material appearance from regions labeled by Z in target photo Y onto regions labeled by X in the input material maps. We show examples of how to define the label in Fig. 4. The sliced Wasserstein loss will be computed between deep features on each labeled regions. Without loss of generality, we extract a deep feature p^l and \hat{p}^l from layer l on the rendered image $R(\theta)$ and one of the material target \hat{I} . Similar to [HVCB21], we randomly sample N directions $V \in S^{N_i}$ and project features p^l and \hat{p}^l onto the sampled directions to get projected 1D features p_V^l and \hat{p}_V^l .

Now suppose we have a transfer rule **Label i: Target \hat{I} , j**, instructing us to transfer materials from regions labeled by j in \hat{I} to regions labeled by i in the rendered image $R(\theta)$. We take samples labeled with i from p_V^l and samples labeled with j from \hat{p}_V^l as $p_V^l\{i\}$ and $\hat{p}_V^l\{j\}$. Note here $p_V^l\{i\}$ and $\hat{p}_V^l\{j\}$ usually contain different number of samples, therefore we compute the sliced Wasserstein loss using our proposed resampling technique (Eq. 4). We compute this loss for each transfer rule separately and take their average as our final loss. For completeness, we also evaluate the Gram Matrix [GEB*17] for partial transfer and show that it can lead to artifacts as shown in Fig. 6. We therefore adopt sliced Wasserstein loss with resampling in all our experiments.

A particular case is made of the boundary features, as neurons on the labeled boundary will have a receptive field which crosses the boundary due to the footprint of the deep convolutions, forcing them to consider statistics from irrelevant nearby regions. To prevent transferring unrelated material statistics, similar to Gatys et al [GEB*17], we perform an erosion operation on the labeled regions, and only evaluate the sliced Wasserstein loss on the eroded regions. Fig. 5 shows an example with and without this erosion step. We note that while an erosion step reduces irrelevant texture transfer, too large an erosion may remove all samples from the distribution at deeper layers. In such case, we do not compute the loss for deeper layers with no valid pixels.

4. Implementation Details

We implement our algorithm in PyTorch for both training and optimization. We train our tileable MaterialGAN model using a synthetic material dataset containing 199,068 images with a resolution of 288x288 [DAD*18]. The material maps are encoded as 9-channel 2D images (3 for albedo, 2 for normal, 1 for roughness and 3 for specular). The full model is trained by crops: we train the generator to synthesize 512x512 material maps and we make a 2x2 tile (1024x1024) and randomly crop to 256x256 to compare with the randomly cropped 256x256 ground truth material maps.

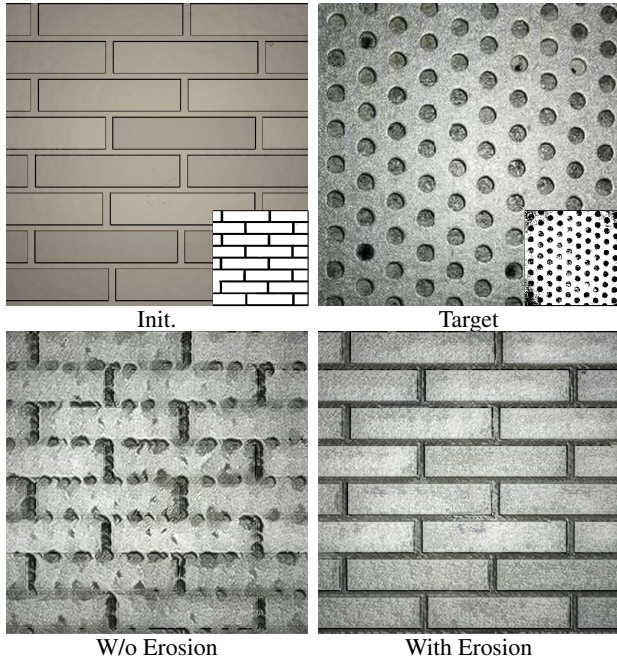


Figure 5: We show the importance of applying erosion on label maps. When computing appearance features at boundary pixels of the label map, the receptive field extends beyond the boundary. Therefore without erosion, the transferred appearance would include regions outside the label mask. Erosion of the binary masks makes sure the appearance features only include information inside the masked region, preserving the desired appearance.

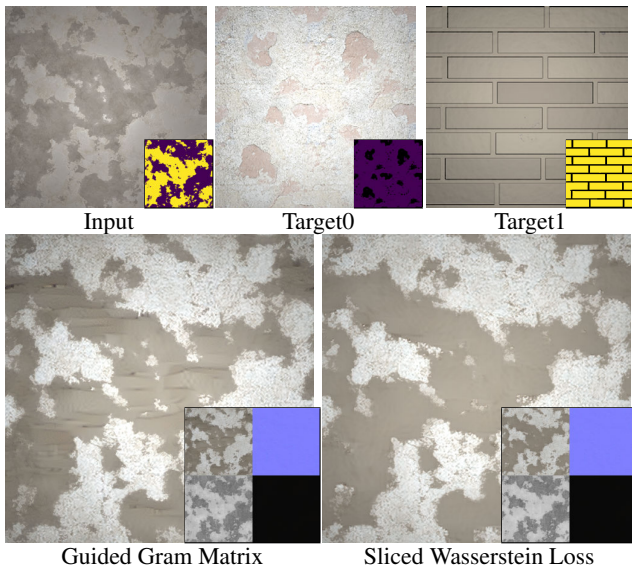


Figure 6: For our multi-target transfer, we evaluate a Guided Gram Matrix which leads to artifacts which do not appear with our re-sampled Wasserstein loss. We hypothesize that these artifacts appear because the Guided Gram Matrix remains impacted by the statistics of target bricks borders despite the erosion we perform.

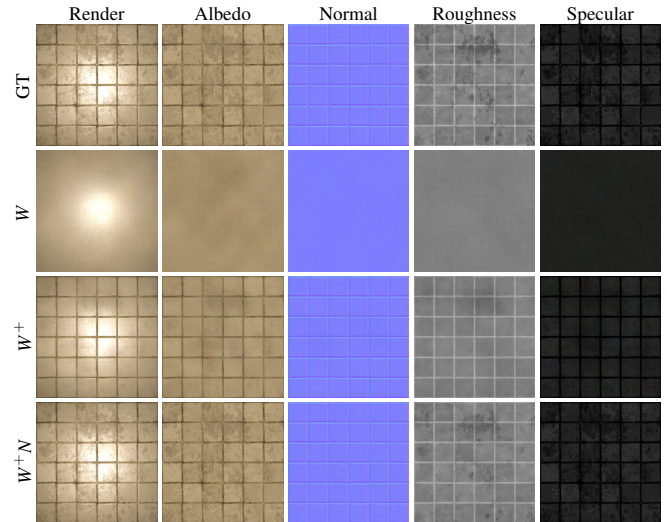


Figure 7: We show projection results when optimizing Eq. 2 with three different latent space of tileable MaterialGAN: W , W^+ and W^+N . We see that we can only capture both large and small scale structures using W^+N , while W , W^+ either fail to approximate the detail appearance or miss fine scale structures.

The architecture of the GAN model ensure tileability (Sec. 3.1), despite the crops being not tileable. For important hyperparameters, we empirically set $\gamma = 10$ for R1 regularization and weight of path length regularization as 1 [KLA*20]. We train the network using an Adam optimizer ($\beta = (0.9, 0.999)$) with a learning rate of 0.002 on 8 Nvidia Tesla V100 GPUs. The full training takes 2~3 days with a batch size of 32.

We run experiments on our optimization-based material transfer on an Intel i9-10850K machine with Nvidia RTX 3090. The optimization is built on the pre-trained tileable MaterialGAN model as a material prior. Specifically, MaterialGAN has multiple latent spaces: $z \in Z$, the input latent code; $w \in W$, the intermediate latent code after linear mapping; per-layer style code $w^+ \in W^+$; and noise inputs for each blocks $n \in N$. In previous work, different latent codes have been proposed for various applications [KAH*20, GSH*20]. In our experiments, we optimize both W^+ and N , enabling our optimization to capture both large-scale structure and fine-scale details. In Fig. 7, we show projection results (Eq. 2) only by optimizing W or W^+ . Results show that W is less expressive, while W^+ can capture large scale structures, but misses fine scale structures.

For the projection step, we run 1000 iterations with an Adam optimizer with a learning rate of 0.08, taking around 5 minutes. We extract deep features from [relu1_2, relu2_2, relu3_2, relu4_2] in a pre-trained VGG19 neural network to evaluate the feature loss in Eq. 2. As we find the projection of the details in the normal map to be harder than for other maps, we assign a weight of 5 to the normal maps loss while 1 for other material maps.

After projection, we optimize the embedded latent code θ to minimize the loss function in Eq. 3. Similar to style transfer, we take deep features from relu4_2 to compute feature loss, and ex-

tract deep features from layers [relu1_1, relu2_1, relu3_1, relu4_1] to compute the sliced Wasserstein loss. We weigh style losses from different layers by [5, 5, 5, 0.5] respectively, emphasizing local features. To compute the sliced Wasserstein loss, we sample a number of random projection directions equal to the number of channels of the compared deep features as suggested in the original paper [HVCB21]. We run 500 iterations using an Adam optimizer with a learning rate of 0.02, taking about 2.5 minutes.

For spatial control, if the input material maps come from a procedural material [Ado22b], the label maps can be extracted from the graph. For target photos, we compute the label maps with a scribble-based segmentation method [HHD*22], which takes less than a minute overall. However, a precise and full segmentation of the example photos is not always necessary: users only need to indicate a few example regions of the material they want to transfer. We apply an erosion operation on each resampled label map with a kernel size of 5 pixels for 512x512 images as described in Sec. 3.3. This erosion size may need to be adapted for different image resolutions.

5. Results

We present single-target material transferred results including material maps and tiled renders in Fig. 8, showing that our method can faithfully transfer realistic details from real photographs to material maps thanks to material prior provided by our modified MaterialGAN, preserving the tileability of input materials. We also show multi-target transfer results in Fig. 9 where we are able to transfer material appearance from separate regions from multiple sources. As with a single target, the new material maps preserve tileability allowing to directly use them for texturing. Please see our supplemental material for more results with dynamic lighting.

5.1. Applications

Using our material transfer method, we demonstrate different applications:

Material Augmentation. Our method can augment existing material maps based on input photos. This is particularly useful for augmenting procedural materials which are difficult to design realistically. As shown in Fig. 10, our method can be applied to minimize the gap between an unrealistic procedural material and a realistic photo, which can be used as an complementary method for existing inverse procedural material modeling systems [HDR19, SLH*20].

By-example Scenes Design. As our approach ensures tileability after optimization, the transferred material maps can be directly applied to texture virtual scenes smoothly. Fig. 1 shows such an application scenario. Given photographs as exemplars, our method transfers realistic details onto simple-looking materials. With our transferred material maps, we can texture and render the entire scene seamlessly.

5.2. Comparison

As we cannot compare to traditional style transfer methods since they operate on the image to image domain, we evaluate the benefit

of the material prior used in our optimization and compare to two alternative optimization approaches, evaluating a simple per-pixel image style transfer combined with differentiable rendering and using a deep image prior [UVL18]. We also evaluate a direct single image material acquisition using MaterialGAN on target images and show that the input material regularizes the result.

5.2.1. Per-pixel Optimization and Deep Image Prior

As discussed in Sec. 3.1, directly performing per-pixel optimization on material maps leads to numerous local minima, resulting in artifacts. The optimization is also very sensitive to initialization and learning rate (Fig. 2), and loss is prone to diverge if the hyperparameters are not tuned well.

Deep Image prior [UVL18] is another way to regularize optimization. The image is reparameterized by a neural network, helping to overcome potential local minima. We adapt this idea to our material transfer optimization. As described in the original paper, we use a U-net-like architecture to generate a stacked 9-channel materials map, using a 9-channel set of randomly initialized noise maps as input. During the optimization, the parameters of this network will be optimized while the input noise map is fixed. Different from the original paper, we do not optimize from scratch but first fit the generator network to our input material maps using Eq. 2, otherwise the optimization cannot recover the structure of the input material maps. After fitting the neural network, we perform the material transfer as described in Eq. 3. Fig. 12 shows an optimization result generated using this deep image prior version. The neural network prior helps address the local minimum problem encountered in the per-pixel optimization, but its lack of prior on tileable materials results in artifacts and does not preserve tileability. In particular we see in Fig. 12 that the result looks good with a frontal lighting, but shows significant green coloration when tiled or with varying light.

5.2.2. Comparison to Material Acquisition

Our material transfer algorithm is not a material acquisition method because we do not aim at faithfully reconstructing the per-pixel material properties of a single material, but rather to transfer appearance statistics of one or more photographs to the input maps. However, as shown in Fig. 11, compared to a per-pixel material acquisition approach, our optimization is regularized by the initialization with the original material, preventing it from overfitting to a single light/view configuration.

6. Discussions, Limitations and Future Work

Our material transfer provides an efficient approach to control and augment appearance of material maps, showing good transfer results in different scenarios with various targets. Our method however fails to find implicit correspondences between input material maps and target photograph in the case of very different patterns or scales, without explicit spatial guidance (Sec. 3.3), as shown in Fig. 13a. In case of strongly conflicting material properties in the source material and target photo(s), our method tends to mix them. As shown in Fig. 13b, if we transfer a photo of a rough marble texture to a pure metallic material, our method mixes their material parameters.

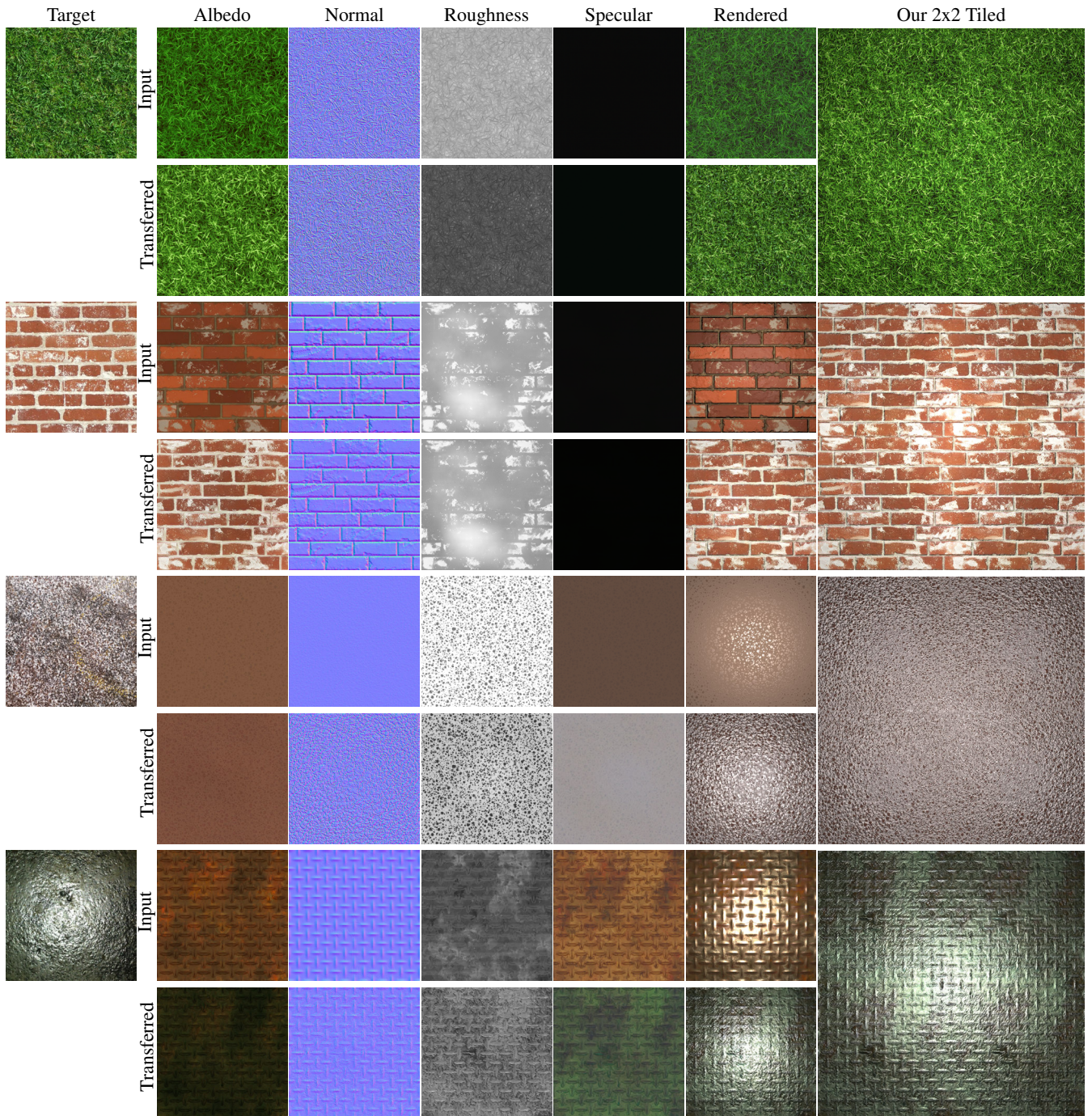


Figure 8: Material Transfer Results. All target images are real photographs. We show each material map (512x512) before and after optimization. We also show a 2x2 tiled rendered image (1024x1024) using our transferred material maps. The optimized material maps have the same large-scale structure as the input material maps but now share the material appearance of the target photo. We see that our transfer does not only modify the diffuse albedo, but can modify various properties when required.

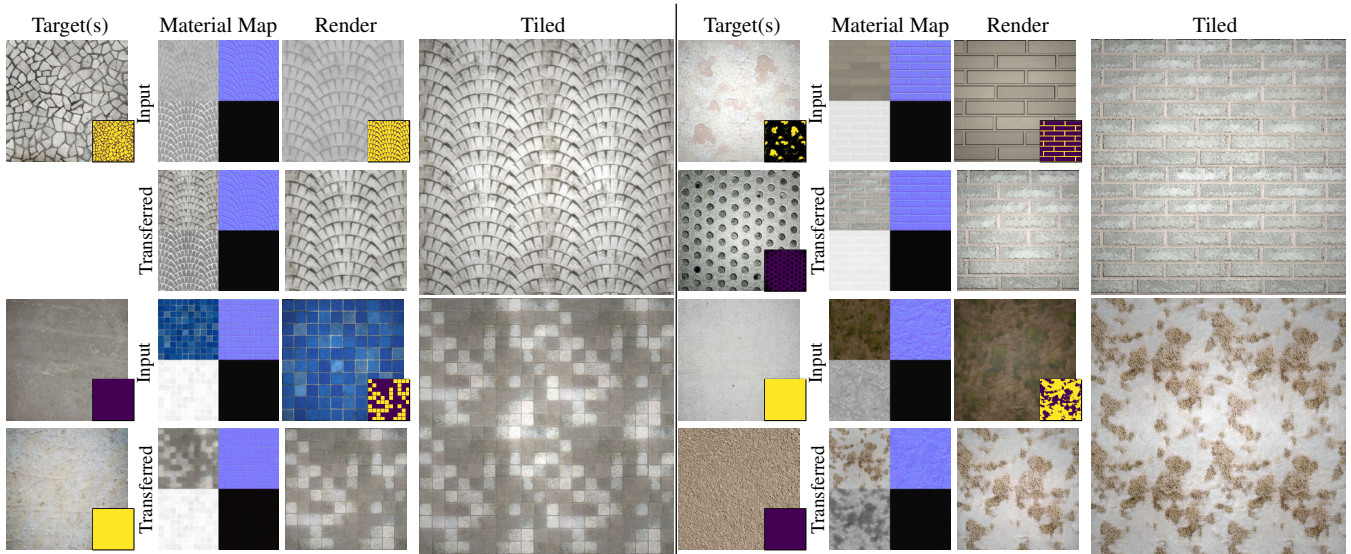


Figure 9: Multi-target material transfer results of our algorithm from regions in multiple targets to regions in the input material. The newly created material maps remain tileable and can be directly used for content creation. The inset color label maps show the transfer correspondence: we transfer material appearance from and onto regions with same color.

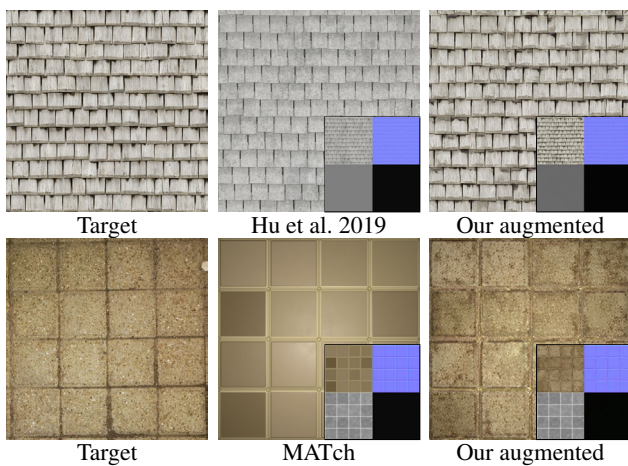


Figure 10: Augmenting inverse procedural results. The procedural materials shown here (middle column) are generated by inverse procedural modeling method [HDR19, SLH*20], but depending on the expressivity of the procedural models, the fitted/optimized material often looks unrealistic. Our method can be applied to improve the realism and augment details of procedural materials generated maps, to better match a target appearance.

As shown in Fig. 14a, to roughly control the impact of the of statistics we want to transfer, we can adjust weights between feature loss d_0 and style loss d_1 . Precise control of the scale at which transfer happens is more challenging. In Fig. 14b, we experiment with transferring statistics from a single VGG layer, showing the different levels of transfer. This however doesn't provide precise transfer scale control. Empirically, we combine statistics from multiple layers (Sec. 4) to produce high-quality transfer.

Another limitation of the method comes from the use of a tileable

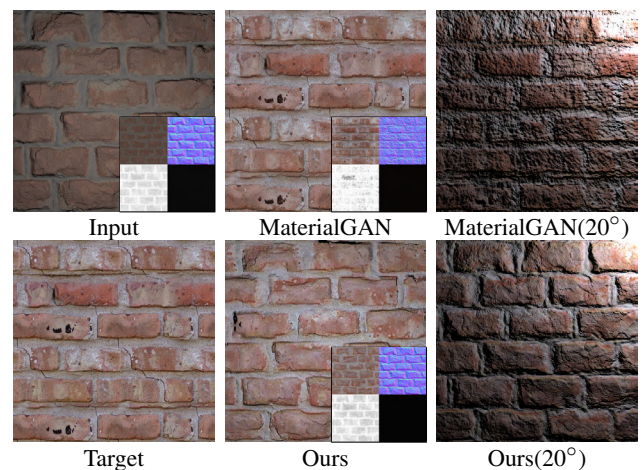


Figure 11: While we do not aim at reconstructing exact per-pixel material parameters given a photograph, direct single acquisition method could be used when the input material structure is not important. MaterialGAN [GSH*20] achieves high quality per-pixel result on the optimized view but over-fit to it, leading to heavy quality loss under other light-view settings (20°). Our approach does not suffer from this thanks to the regularization provided by the input material.

MaterialGAN which is trained at a fixed resolution. Though its base architecture, StyleGAN, has shown great results up to 1K resolution images, the model does not yet trivially support super-high resolution materials used in large entertainment productions (4K to 8K).

Finally, we use a Cook-Torrance-like material representation, limiting the materials that can be modelled to opaque surfaces,

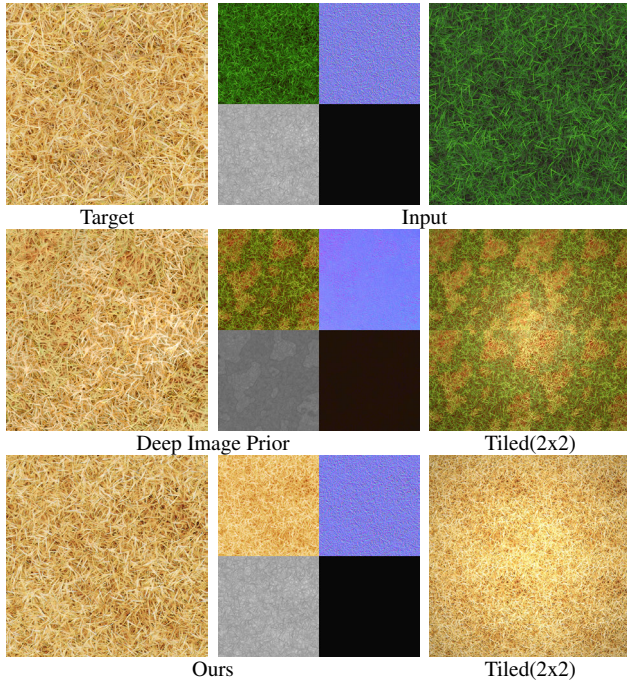


Figure 12: We compare our MaterialGAN prior with a material transfer based on Deep Image Prior [UVL18]. Compared to a per-pixel optimization, the Deep Image Prior helps avoid local minima to reach the target appearance. However, without material prior and explicit tileable design, Deep Image Prior fails to preserve tileability and to generate view/lighting consistent result. This is illustrated here by the remaining green grass appearing when lighting is not exactly the same as during the optimization.

and requiring example photographs under roughly known conditions (e.g flash or sun). An interesting future direction is to explore more complex material effects and allowing material transfer from non-planar in-the-wild objects example.

7. Conclusion

We design a novel algorithm to control the appearance of 2D material maps through material appearance transfer. For high-quality material transfer, we train a tileable MaterialGAN, leveraging its learned space as an optimization prior and differentiable rendering to use simple photographs as target appearance. We introduce spatial control with multi-target transfer using a resampled sliced Wasserstein loss and show complex by-example control and augmentation. The newly-synthesized material maps can be used seamlessly in any virtual environment. We believe our approach provides users with a new effective and convenient way to control the appearance of material maps and create new materials, improving the toolbox for virtual content creation.

Acknowledgment

This work was supported in part by NSF Grant No. IIS-2007283.

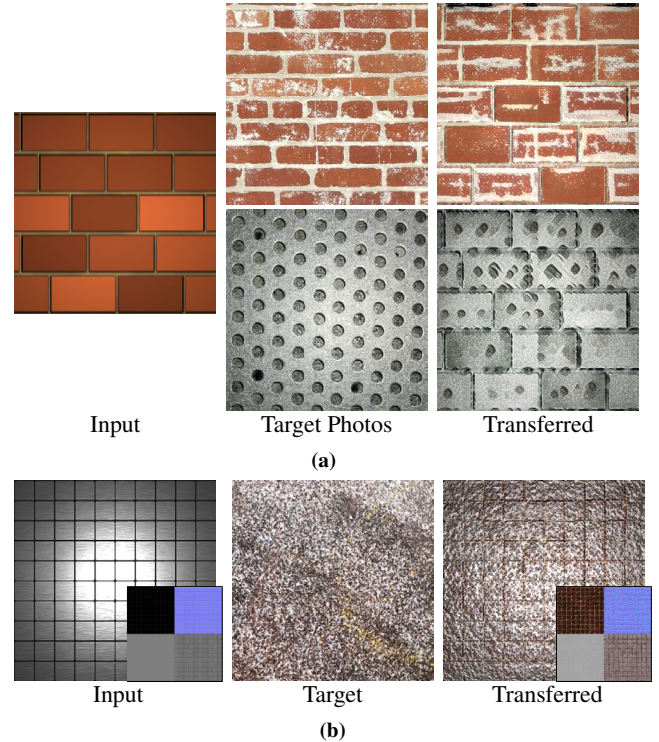


Figure 13: Limitation: (a) when input material maps and target photos have very different structures or scale, without spatial guidance (label maps), our method cannot find a good way to transfer the high frequency statistics. (b) In the case where the target photo and input material maps have strongly conflicting material properties (here roughness), our method will result in mixed parameters.

References

- [AAL16] AITTA M., AILA T., LEHTINEN J.: Reflectance modeling by neural texture synthesis. *ACM Transactions on Graphics (ToG)* 35, 4 (2016), 1–13. 3
- [Ado22a] ADOBE: Adobe Photoshop, 2022. 1
- [Ado22b] ADOBE: Substance Designer, 2022. 2, 7
- [AP08] AN X., PELLACINI F.: AppProp: All-Pairs Appearance-Space Edit Propagation. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, Association for Computing Machinery. doi:10.1145/1399504.1360639. 2
- [BAEDR08] BEN-ARTZI A., EGAN K., DURAND F., RAMAMOORTHY R.: A Precomputed Polynomial Representation for Interactive BRDF Editing with Global Illumination. *ACM Trans. Graph.* 27, 2 (may 2008). doi:10.1145/1356682.1356686. 2
- [BAOR06] BEN-ARTZI A., OVERBECK R., RAMAMOORTHY R.: Real-Time BRDF Editing in Complex Lighting. *ACM Trans. Graph.* 25, 3 (jul 2006), 945–954. doi:10.1145/1141911.1141979. 2
- [Cra28] CRAMÉR H.: *On the composition of elementary errors: Statistical applications*. Almqvist and Wiksell, 1928. 2
- [CT82] COOK R. L., TORRANCE K. E.: A Reflectance Model for Computer Graphics. *ACM Trans. Graph.* 1, 1 (jan 1982), 7–24. doi:10.1145/357290.357293. 2
- [DAD*18] DESCHAINTE V., AITTA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Single-Image SVBRDF Capture with a Rendering-Aware Deep Network. *ACM Trans. Graph.* 37, 4 (Aug 2018). 1, 3, 5

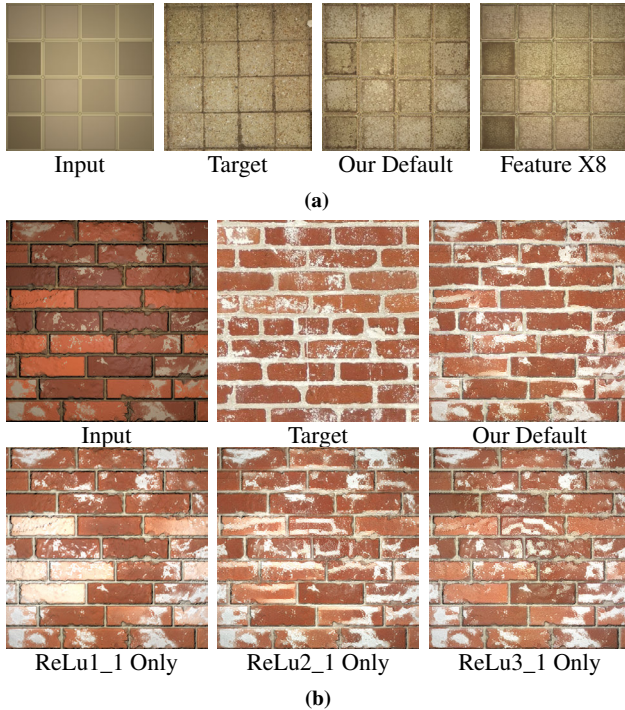


Figure 14: (a) Adjusting the relative strength of feature loss and style loss impacts how much statistics is transferred to the source material. As we increase the relative weight of feature loss ($\times 8$), the output will look more like the original input. (b) We combine statistics from multiple deep layers of VGG19 [SZ15] for a good micro- and meso-scale transfer (our default). Transferring specific VGG layers (ReLU*_1 Only) results in different scale of transfer, but doesn't provide precise control.

[DAD*19] DESCHAINTE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Flexible SVBRDF Capture with a Multi-Image Deep Network. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 38, 4 (July 2019), 1–13. 1, 3

[DDB20] DESCHAINTE V., DRETTAKIS G., BOUSSEAU A.: Guided Fine-Tuning for Large-Scale Material Transfer. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 39, 4 (2020), 91–105. URL: <http://www-sop.inria.fr/reves/Basilic/2020/DDB20>. 2, 3

[FJL*16] FIŠER J., JAMRIŠKA O., LUKÁČ M., SHECHTMAN E., ASENTE P., LU J., SÝKORA D.: StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings. *ACM Trans. Graph.* 35, 4 (jul 2016). doi:10.1145/2897824.2925948. 2

[GEB15] GATYS L., ECKER A. S., BETHGE M.: Texture Synthesis using Convolutional Neural Networks. *Advances in neural information processing systems* 28 (2015). 3, 4

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image Style Transfer using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2414–2423. 2, 3

[GEB*17] GATYS L. A., ECKER A. S., BETHGE M., HERTZMANN A., SHECHTMAN E.: Controlling Perceptual Factors in Neural Style Transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 3985–3993. 5

[GGG*16] GUARNERA D., GUARNERA G. C., GHOSH A., DENK C.,

GLENCROSS M.: BRDF Representation and Acquisition. *Computer Graphics Forum* 35, 2 (2016), 625–650. 3

[GHYZ20] GUO Y., HAŠAN M., YAN L., ZHAO S.: A Bayesian Inference Framework for Procedural Material Parameter Estimation. *Computer Graphics Forum* 39, 7 (2020), 255 – 266. 3

[GLD*19] GAO D., LI X., DONG Y., PEERS P., XU K., TONG X.: Deep Inverse Rendering for High-resolution SVBRDF Estimation from an Arbitrary Number of Images. *ACM Trans. Graph.* 38, 4 (July 2019). doi:10.1145/3306346.3323042. 1, 3

[GLT*21] GUO J., LAI S., TAO C., CAI Y., WANG L., GUO Y., YAN L.-Q.: Highlight-Aware Two-Stream Network for Single-Image SVBRDF Acquisition. *ACM Trans. Graph.* 40, 4 (July 2021). doi:10.1145/3450626.3459854. 3

[GMD*16] GRYADITSKAYA Y., MASIA B., DIDYK P., MYSZKOWSKI K., SEIDEL H.-P.: Gloss Editing in Light Fields. In *Proceedings of the Conference on Vision, Modeling and Visualization (Goslar Germany, Germany, 2016)*, VMV '16, Eurographics Association, pp. 127–135. doi:10.2312/vmv.20161351. 2

[GSH*20] GUO Y., SMITH C., HAŠAN M., SUNKAVALLI K., ZHAO S.: MaterialGAN: Reflectance Capture Using a Generative SVBRDF Model. *ACM Trans. Graph.* 39, 6 (Nov. 2020). doi:10.1145/3414685.3417779. 1, 2, 3, 6, 9

[HB17] HUANG X., BELONGIE S.: Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 1501–1510. 3

[HDMR21] HENZLER P., DESCHAINTE V., MITRA N. J., RITSCHEL T.: Generative Modelling of BRDF Textures from Flash Images. *ACM Trans Graph (Proc. SIGGRAPH Asia)* 40, 6 (2021). 3

[HDR19] HU Y., DORSEY J., RUSHMEIER H.: A Novel Framework for Inverse Procedural Texture Modeling. *ACM Trans. Graph.* 38, 6 (Nov. 2019). doi:10.1145/3355089.3356516. 2, 3, 4, 7, 9

[HGC*20] HU B., GUO J., CHEN Y., LI M., GUO Y.: DeepBRDF: A Deep Representation for Manipulating Measured BRDF. *Computer Graphics Forum* (2020). doi:10.1111/cgf.13920. 2

[HHD*22] HU Y., HE C., DESCHAINTE V., DORSEY J., RUSHMEIER H.: An Inverse Procedural Modeling Pipeline for SVBRDF Maps. *ACM Trans. Graph.* 41, 2 (jan 2022). doi:10.1145/3502431. 2, 3, 7

[HVCB21] HEITZ E., VANHOEY K., CHAMBON T., BELCOUR L.: A Sliced Wasserstein Loss for Neural Texture Synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021). 2, 3, 4, 5, 7

[HZL19] HUANG Z., ZHANG J., LIAO J.: Style Mixer: Semantic-aware Multi-Style Transfer Network. *Computer Graphics Forum* (2019). doi:10.1111/cgf.13853. 3

[JAFF16] JOHNSON J., ALAHI A., FEI-FEI L.: Perceptual Losses for Real-time Style Transfer and Super-resolution. In *European conference on computer vision* (2016), Springer, pp. 694–711. 3

[JvST*19] JAMRIŠKA O., ŠÁRKA SOCHOROVÁ, TEXLER O., LUKÁČ M., FIŠER J., LU J., SHECHTMAN E., SÝKORA D.: Stylizing Video by Example. *ACM Transactions on Graphics* 38, 4 (2019). 3

[KAH*20] KARRAS T., AITTALA M., HELLSTEN J., LAINE S., LEHTINEN J., AILA T.: Training Generative Adversarial Networks with Limited Fata. *Advances in Neural Information Processing Systems* 33 (2020), 12104–12114. 6

[KAL*21] KARRAS T., AITTALA M., LAINE S., HÄRKÖNEN E., HELLSTEN J., LEHTINEN J., AILA T.: Alias-Free Generative Adversarial Networks. In *Proc. NeurIPS* (2021). 3

[KLA*20] KARRAS T., LAINE S., AITTALA M., HELLSTEN J., LEHTINEN J., AILA T.: Analyzing and Improving the Image Quality of Stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 8110–8119. 3, 4, 6

[KSS19] KOLKIN N., SALAVON J., SHAKHAROVICH G.: Style Transfer by Relaxed Optimal Transport and Self-Similarity. In *Proceedings of*

- the *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 3
- [LBAD*06] LAWRENCE J., BEN-ARTZI A., DECORO C., MATUSIK W., PFISTER H., RAMAMOORTHY R., RUSINKIEWICZ S.: Inverse Shade Trees for Non-Parametric Material Representation and Editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 25, 3 (July 2006). 2
- [LDPT17] LI X., DONG Y., PEERS P., TONG X.: Modeling Surface Appearance from a Single Photograph using Self-augmented Convolutional Neural Networks. *ACM Trans. Graph.* 36, 4 (2017). 3
- [LL11] LEPAGE D., LAWRENCE J.: Material Matting. *ACM Trans. Graph.* 30, 6 (dec 2011), 1–10. doi:10.1145/2070781.2024178. 2
- [MMTK19] MAZLOV I., MERZBACH S., TRUNZ E., KLEIN R.: Neural Appearance Synthesis and Transfer. In *Workshop on Material Appearance Modeling* (2019), Klein R., Rushmeier H., (Eds.), The Eurographics Association. doi:10.2312/mam.20191311. 3
- [NDM06] NGAN A., DURAND F., MATUSIK W.: Image-Driven Navigation of Analytical BRDF Models. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques* (Goslar, DEU, 2006), EGSR '06, Eurographics Association, p. 399–407. 2
- [NRM*12] NGUYEN C. H., RITSCHEL T., MYSZKOWSKI K., EISEMANN E., SEIDEL H.-P.: 3D Material Style Transfer. *Computer Graphics Forum (Proc. EUROGRAPHICS 2012)* 2, 31 (2012). 3
- [Pho75] PHONG B. T.: Illumination for Computer Generated Pictures. *Commun. ACM* 18, 6 (jun 1975), 311–317. doi:10.1145/360825.360839. 2
- [RPG22] RODRIGUEZ-PARDO C., GARCÉS E.: Neural Photometry-guided Visual Attribute Transfer. *IEEE Transactions on Visualization and Computer Graphics* (2022). 2
- [SGM*16] SERRANO A., GUTIERREZ D., MYSZKOWSKI K., SEIDEL H.-P., MASIA B.: An Intuitive Control Space for Material Appearance. *ACM Trans. Graph.* 35, 6 (nov 2016). doi:10.1145/2980179.2980242. 2
- [SJT*19] SÝKORA D., JAMRIŠKA O., TEXLER O., FIŠER J., LUKÁČ M., LU J., SHECHTMAN E.: StyleBlit: Fast Example-Based Stylization with Local Guidance. *Computer Graphics Forum* 38, 2 (2019), 83–91. 3
- [SLH*20] SHI L., LI B., HAŠAN M., SUNKAVALLI K., BOUBEKEUR T., MECH R., MATUSIK W.: MATch: Differentiable Material Graphs for Procedural Material Capture. *ACM Trans. Graph.* 39, 6 (Dec. 2020). 2, 3, 7, 9
- [SWSR21] SHI W., WANG Z., SOLER C., RUSHMEIER H.: A Low-Dimensional Perceptual Space for Intuitive BRDF Editing. In *EGSR 2021 - Eurographics Symposium on Rendering - DL-only Track* (Saarbrücken, Germany, June 2021), pp. 1–13. URL: <https://hal.inria.fr/hal-03364272>. 2
- [SZ15] SIMONYAN K., ZISSERMAN A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015), Bengio Y., LeCun Y., (Eds.). URL: <http://arxiv.org/abs/1409.1556>. 3, 4, 11
- [TFF*20] TEXLER O., FUTSCHIK D., FIŠER J., LUKÁČ M., LU J., SHECHTMAN E., SÝKORA D.: Arbitrary Style Transfer Using Neurally-Guided Patch-Based Synthesis. 3
- [TFK*20] TEXLER O., FUTSCHIK D., KUČERA M., JAMRIŠKA O., ŠÁRKA SOCHOROVÁ, CHAI M., TULYAKOV S., SÝKORA D.: Interactive Video Stylization Using Few-Shot Patch-Based Training. *ACM Transactions on Graphics* 39, 4 (2020), 73. 3
- [TGY*09] TALTON J. O., GIBSON D., YANG L., HANRAHAN P., KOLTUN V.: Exploratory Modeling with Collaborative Design Spaces. *ACM Trans. Graph.* 28, 5 (dec 2009), 1–10. doi:10.1145/1618452.1618513. 2
- [TTK*21] TEXLER A., TEXLER O., KUČERA M., CHAI M., SÝKORA D.: FaceBlit: Instant Real-time Example-based Style Transfer to Facial Videos. *Proceedings of the ACM in Computer Graphics and Interactive Techniques* 4, 1 (2021), 14. 3
- [UVLV16] ULYANOV D., LEBEDEV V., VEDALDI A., LEMPITSKY V.: Texture Networks: Feed-Forward Synthesis of Textures and Stylized Images. *ICML'16, JMLR.org*, p. 1349–1357. 3
- [UVL18] ULYANOV D., VEDALDI A., LEMPITSKY V.: Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 9446–9454. 7, 10
- [WAKB09] WILLS J., AGARWAL S., KRIEGMAN D., BELONGIE S.: Toward a Perceptual Space for Gloss. *ACM Trans. Graph.* 28, 4 (sep 2009). doi:10.1145/1559755.1559760. 2
- [WMLT07a] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet Models for Refraction through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Goslar, DEU, 2007), EGSR'07, Eurographics Association, p. 195–206. 2
- [WMLT07b] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet Models for Refraction through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Goslar, DEU, 2007), EGSR'07, Eurographics Association, p. 195–206. 3
- [YDPG21] YE W., DONG Y., PEERS P., GUO B.: Deep Reflectance Scanning: Recovering Spatially-varying Material Appearance from a Flash-lit Video Sequence. *Computer Graphics Forum* (2021). doi: <https://doi.org/10.1111/cgf.14387>. 3
- [ZFWW20] ZSOLNAI-FEHÉR K., WONKA P., WIMMER M.: Photorealistic Material Editing Through Direct Image Manipulation. *Computer Graphics Forum* (2020). doi:10.1111/cgf.14057. 2
- [ZHD*22] ZHOU X., HAŠAN M., DESCHAINTE V., GUERRERO P., SUNKAVALLI K., KALANTARI N.: Tilegen: Tileable, controllable material generation and capture, 2022. URL: <https://arxiv.org/abs/2206.05649>. 3, 4
- [ZK21] ZHOU X., KALANTARI N. K.: Adversarial Single-Image SVBRDF Estimation with Hybrid Training. *Computer Graphics Forum* 40, 2 (2021), 315–325. doi: <https://doi.org/10.1111/cgf.142635>. 3
- [ZZB*18] ZHOU Y., ZHU Z., BAI X., LISCHINSKI D., COHEN-OR D., HUANG H.: Non-stationary Texture Synthesis by Adversarial Expansion. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 37, 4 (2018). 3