Eurographics Symposium on Rendering 2024
E Garces and E. Haines
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 4

MatUp: Repurposing Image Upsamplers for SVBRDFs

A. Gauthier[1,3], B. Kerbl[2] J. Levallois[3], R. Faury[3], J. M. Thiery[3], T. Boubekeur[3]

[1]Inria, Université Côte d'Azur, France
[2]TU Wien, Austria
[3]Adobe Research, France

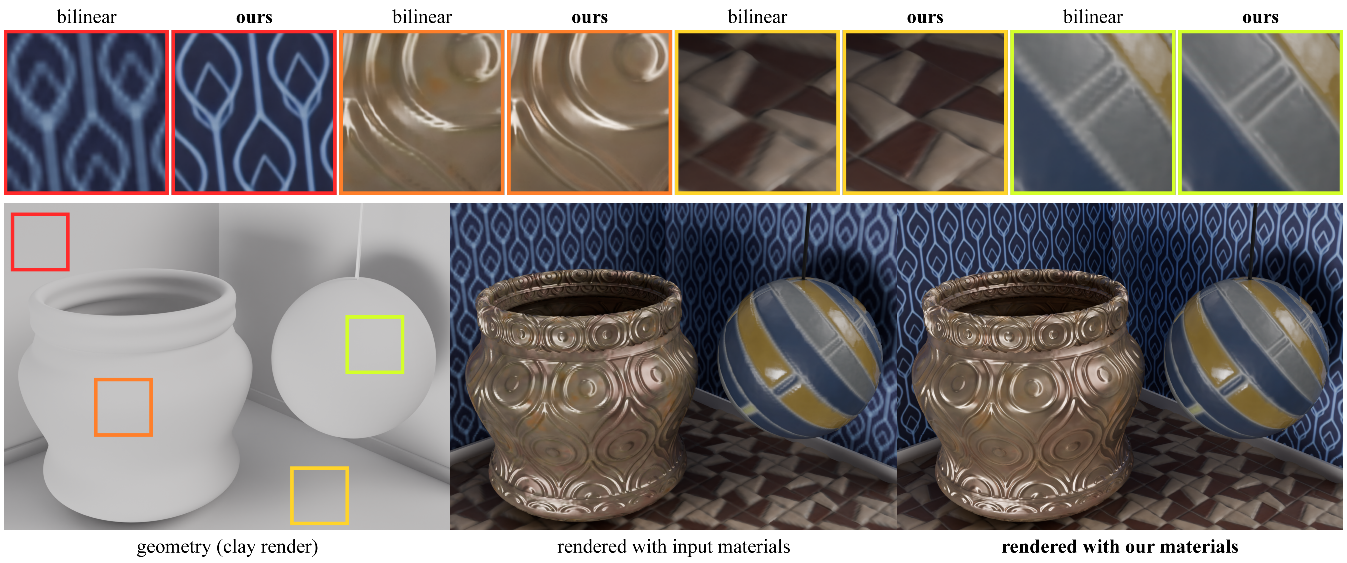


Figure 1: *Given a low-resolution 128² input materials, our method produces 512² outputs preserving tileability and displaying sharper material structures compared to widely-used per-channel upsampling filters (e.g. bilinear or Lanczos) in less than a minute.*

Abstract

We propose MatUp*, an upsampling filter for material super-resolution. Our method takes as input a low-resolution SVBRDF and upscales its maps so that their rendering under various lighting conditions fits upsampled renderings inferred in the radiance domain with pre-trained RGB upsamplers. We formulate our local filter as a compact Multilayer Perceptron (MLP), which acts on a small window of the input SVBRDF and is optimized using a data-fitting loss defined over upsampled radiance at various locations. This optimization is entirely performed at the scale of a single, independent material. Doing so,* MatUp *leverages the reconstruction capabilities acquired over large collections of natural images by pre-trained RGB models and provides regularization over self-similar structures. In particular, our light-weight neural filter avoids retraining complex architectures from scratch or accessing any large collection of low/high resolution material pairs – which do not actually exist at the scale RGB upsamplers are trained with. As a result,* MatUp *provides fine and coherent details in the upscaled material maps, as shown in the extensive evaluation we provide.*

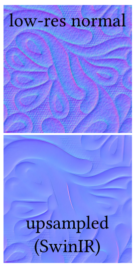
CCS Concepts
• ***Computing methodologies*** → ***Reflectance modeling; Texturing;***

1. Introduction

Digital materials describe the appearance of virtual 3D shapes under arbitrary lighting and come in the form of spatially-varying bidirectional reflectance distributions (SVBRDF) modeled by a small collection of 2D maps. Capturing real-world material samples allows for reproducing their appearance in graphics applications, but is limited by acquisition conditions, which include the physical sample size and distance to the sensor. Generative techniques for materials [GSH*20, HHG*22, ZHD*22, ZHD*23] allow for faster design and exploration but cannot produce ultra high-resolution materials (4K) [HHG*22, ZHD*23], since training for such materials is computationally intensive. Consequently, SVBRDF super-resolution from low-resolution inputs appears as a key operator for both captured and generated materials.

State-of-the-art image super-resolution leverage deep-learning architectures and large RGB image datasets, which do not exist for materials. Although reusing existing trained models seems tempting, naively applying them to SVBRDF maps fails because their color statistics strongly deviate from those of natural RGB images (see Fig. 2). Hence, considering the lack of material data to train from and the immense compute requirements, we propose to reuse existing RGB super-resolution techniques for upsampling the *radiance* of materials as perceived when rendered under a large set of lighting conditions, before upscaling the underlying reflectance maps so that their rendering matches such upsampled radiances. To this end, we introduce MatUp, a neural SVBRDF upsampling filter, optimized to upscale the maps. MatUp targets no reference super-resolution and leverages the self-similarity of typical materials to provide regularization.

Our key initial observation is that RGB-upsamplers provide pleasing results when applied on rendered materials: they add sharp features while retaining the original appearance of the material. However, independently upsampling multiple renderings with varying light conditions leads to inconsistencies (see Fig. 2,a). Alternatively, upscaling each map prior to rendering, as shown in Fig. 2 (b), results in spatially-uncorrelated output maps (see the upper-part of the wooden material, where albedo, roughness and normal do not match spatially), as well as per-channel artifacts (see inset) due mainly to RGB upsamplers being trained on photographs and not scalar height, normal or roughness maps. The best-in-class RGB upsamplers optimize indeed for human-perception-based metrics. While it makes sense to use such metrics for images as the goal is to offer high-resolution images that eventually please the human eye, those metrics are not appropriate for SVBRDF channels such as normal or roughness, as those maps are merely material parameters for radiance computational models.



Instead, we propose to leverage multiple upsampled renderings to recover the SVBRDF parameters using a data-fitting optimization. This has two main advantages: we can use pre-trained RGB super-resolution algorithms to provide data for optimization and leverage the input low-resolution SVBRDF as a prior for estimating SVBRDF parameters in a highly-controlled lighting and shading environment. We design our algorithm with the following goals in mind: (i) lightweight to enable rapid optimization (ii) structure-preserving so that tileable inputs produce tileable outputs, (iii) parametric, with the possibility to exploit a variety of RGB upscalers as super resolution is admittedly an ill-posed problem.

2. Related work

Single-image RGB super-resolution Super-resolution of RGB images is an active research topic. In our context, we are provided with a single material (i.e., a single stack of SVBRDF maps)

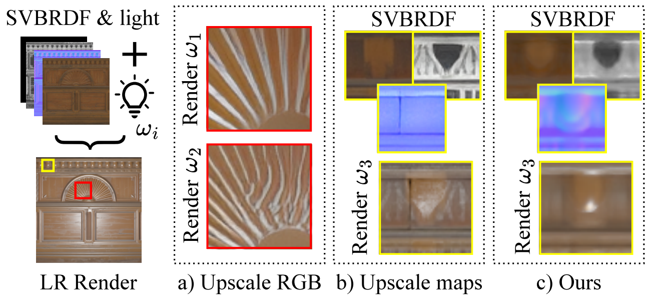


Figure 2: *(Left) Given a Low-Resolution (LR) SVBRDF and a light direction ω_i as input: a) Upscaling LR renders using SwinIR x4 [LCS*21] leads to inconsistencies (renders under ω_1, ω_2). b) Upscaling directly LR SVBRDFs using existing RGB-upsamplers is not appropriate, as these are designed to upscale radiances (or photographs). Notice how the structures do not overlap (SVBRDF and renders ω_3). c) Given an RGB-upsampler $\mathcal{S}$, we optimize for the upscaling of the input material, in order to retain $\mathcal{S}$'s resolution enhancement capabilities in the shading.*

to upsample. For this task, learning-based methods [WYW*19, LCS*21, LDVGT20, WXDS21, SHC*22] have shown their capacity at adding plausible high-frequency details to unseen images, by leveraging datasets of pairs of low- and high-resolution images. They rely on degradation models which include blur, resize, noise and compression operations, which imitate plausible real world inputs [ZLVGT21, WXDS21]. When dealing with SVBRDF, downsampling of parameter maps is non-trivial as these quantities do not have a linear relationship with the final rendered image [BN12]. However, shaded materials create images which imitate natural images, in the sense that radiance captured on a virtual sensor (rendered images) computed from SVBRDF maps models real world material acquisition (bricks, grounds, metals, etc.). Hence, applying super-resolution algorithms to rendered materials provides a solid base for radiance super-resolution. As largely emphasized in the literature [CZY*19, YZT*19, BM20, WCH21, CHQ*22], *data super-resolution is a fundamentally ill-posed problem*. With this in mind, we allow *exploring several super-resolution options, by taking an input RGB-upsampler as meta-parameter*.

SVBRDF manipulation Procedural material estimation [HDR19, GAD*20, SLH*20, HHD*22, HGH*22, LSM23] allow recovering material graphs from rendered materials, to take advantage of their unique capabilities: they allow for material generation at any resolution [SLH*20, HHD*22, LSM23], with precise control over the appearance. Any material could theoretically be upsampled by simply recovering its graph and computing maps at a higher resolution. Unfortunately, these methods do not generalize over all possible inputs and sometimes fail to recover complex mesostructures of in the inputs. Inhomogeneous material synthesis [ZSL*17, MMTK19] allows for generating and expanding RGB or material textures by leveraging exemplars, but do not focus on the super-resolution task. AppIm [DRCP14] allows for editing spatial variations inside SVBRDFs, but does not deal with the aliasing occurring when upsampling each map independently (such as bilinearly upsampled diffuse color in linear space). This framework can however be used

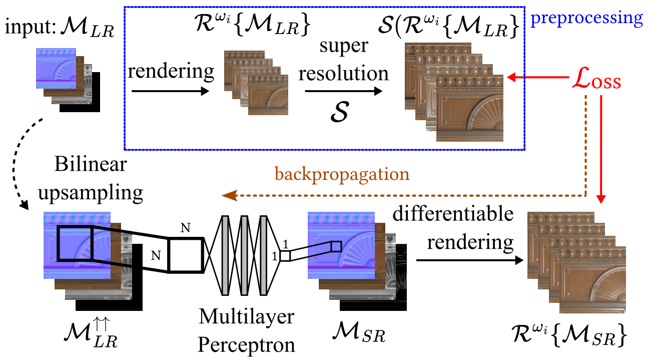

input: $\mathcal{M}_{LR}$
$\mathcal{R}^{\omega_i}\{\mathcal{M}_{LR}\}$
$\mathcal{S}(\mathcal{R}^{\omega_i}\{\mathcal{M}_{LR}\}$
preprocessing
rendering
super resolution $\mathcal{S}$
$\mathcal{L}$oss
Bilinear upsampling
backpropagation
differentiable rendering
N
N
1
1
$\mathcal{M}_{LR}^{\uparrow\uparrow}$
Multilayer Perceptron
$\mathcal{M}_{SR}$
$\mathcal{R}^{\omega_i}\{\mathcal{M}_{SR}\}$

Figure 3: *Overview of our method for radiance-based SVBRDF super-resolution. The main metaparameters of our approach are a shading model $\mathcal{R}$, a super-resolution algorithm $\mathcal{S}$, and a tile size N.*

as a post-process of our method. SVBRDF downsampling was also explored in the literature [OB10, DHI*13, GFL*22], but those approaches cannot be easily reversed to enable upsampling. In particular, [GFL*22] uses a cascade of networks to prevent error accumulation. In our case, we focus on defining a robust data-fitting loss to extract valuable information from many upsampled renderings.

Radiance-based SVBRDF acquisition Recovering SVBRDF maps from radiance data enables lightweight material acquisition. Deep learning approaches were proposed, which take as input a single flash-photograph [AAL16, LDPT17, DAD*18, HDMR21, MRR*22], a collection thereof [DAD*19, GLD*19, GSH*20], or flash-lit videos [YDPG21]. The aim of such techniques is to recover a compact representation for material appearance, based on a few samples from real-life material captures. Our framework shares similarities with these approaches, as we recover SVBRDF parameters from radiance data. However, in our framework we have a precise control over the illumination conditions, the number of samples and the shading model. This allows us to provide as much information as required for recovering the parameters. Hence, we do not need strong priors for our optimization, as modeled by previous learning-based approaches through deep learning frameworks. We show how an outlier-sensitive optimization such as l_1 minimization over all samples provides convincing results for this task.

3. Method

Our approach takes as input a set of low-resolution (LR) SVBRDF maps $\mathcal{M}_{LR}$. It is parameterized by a shading model $\mathcal{R}$ and a super-resolution algorithm $\mathcal{S}$, and outputs a set of high-resolution (HR) SVBRDF maps $\mathcal{M}_{SR}$.

We start by rendering the low-resolution inputs materials under a set of illumination conditions ω_i to obtain low-resolution radiance images $\mathcal{R}^{\omega_i}\{\mathcal{M}_{LR}\}$ before upsampling them with a state-of-the-art RGB upscaler. From these upsampled renderings $\mathcal{S}(\mathcal{R}^{\omega_i}\{\mathcal{M}_{LR}\})$, we reconstruct the upscaled parameter maps of the SVBRDF $\mathcal{M}_{SR}$ i.e., base color, normal, metalness, roughness and height maps (see Fig. 3). To do so, we use a differentiable rendering pipeline based on $\mathcal{R}$ which allows back propagating through a small fully-connected Multilayer Perceptron (MLP) which learns to optimize $\mathcal{M}_{SR}$ for a radiance-based loss $\mathcal{L}$, defined for each pixel p of the high-resolution SVBRDF $\mathcal{M}_{SR}$. Next, we detail our loss term and motivate the need for using a supervised neural upsampling filter.

3.1. Collaborative learning of materials upsampling

Given a collection of texels $\mathcal{P} = \{p\}$ (many texels fetched from the input LR SVBRDF), we aim at collectively minimizing their respective loss functions $\{\mathcal{L}(p)\}_{p\in\mathcal{P}}$. As shown in Fig. 2, while many of the upscaled renderings exhibit common structures, outliers appear (in subregions for some of them) in the form of artifacts, discontinuities, or invalid pixels in the upsampled renderings. We design a 2-level fitting strategy to output a clean upscaled SVBRDF in this context:

(i): *At the pixel level*, we use a data-fitting loss $\mathcal{L}_{\text{fit}}$, that allows retaining a pertinent BRDF signal, even in the presence of such outliers (Subsec. 3.1.1).

(ii): *At the material level*, we leverage an MLP as an implicit regularizer to collaboratively optimize for a function mapping input to output. This guarantees consistent upsampling of similar regions (Subsec. 3.1.2).

3.1.1. Outlier-insensitive data fitting loss

We first propose a pixel-level strategy to retrieve high-resolution SVBRDF by leveraging differentiable rendering. As mentioned earlier, each upsampled rendering $\mathcal{S}(\mathcal{R}^{\omega_i}\{\mathcal{M}_{LR}\})$ contains high-resolution information which may be partly inconsistent *between different illumination conditions* (see Fig. 2, red insets). Hence, we propose to collaboratively leverage all renders to optimize for a single SVBRDF which best explains those while enforcing robustness to outliers. Given $\mathcal{P}$, we introduce the following fitting loss:

$$\mathcal{L}_{\text{fit}} = \sum_{p\in\mathcal{P}} \sum_{\omega_i\in\Omega} \left\| \mathcal{S}(\mathcal{R}^{\omega_i}\{\mathcal{M}_{LR}\}(p)) - \mathcal{R}^{\omega_i}\{\mathcal{M}_{SR}\}(p) \right\|_2 \quad (1)$$

$\mathcal{L}_{\text{fit}}$ is therefore a mixed $l_{2,1}$ norm[†] over all data-fitting terms in $\mathcal{P}$, inducing group sparsity at the *light level*, but remaining sensitive to outliers at the *color channel level*. Note that using an $l_{2,2}$ (or $l_{1,1}$, or any $l_{p,p}$) norm instead would allow rewriting $\mathcal{L}_{\text{fit}}$ as a sum over all (r,g,b) channels separately, thus resulting in three separate and independent optimizations for all three color channels in practice.

Per-pixel optimization alone suffers from several limitations (see Fig. 4).

First, RGB supersamplers cannot process very large SVBRDFs (e.g. 4096^2). In addition, such inputs cannot be optimized concurrently because of the high amount of VRAM required during a gradient descent optimization.

[†] The $l_{p,q}$ norm of bi-indexed elements $\alpha = \{\{\alpha_{i,j}\}_j\}_i$ is defined as $\|\alpha\|_{p,q} = \left(\sum_i \left(\sum_j |\alpha_{i,j}|^p\right)^{\frac{q}{p}}\right)^{\frac{1}{q}}$, see [Kow09] or [BJMO11, section 1.1.1]

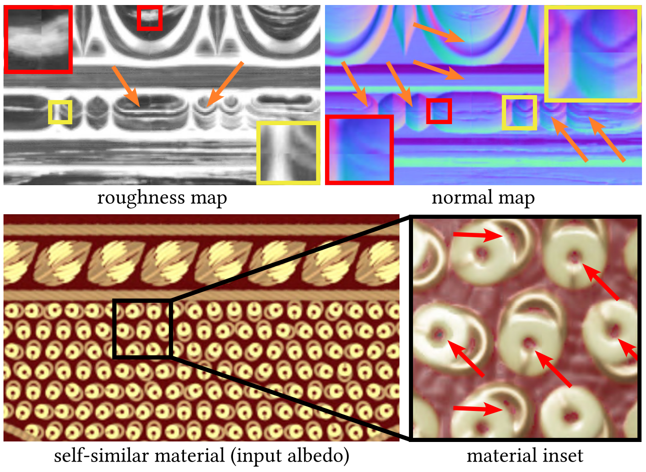


Figure 4: Limitations of a per-pixel approach. Top: *discontinuities between adjacent tiles (orange) in the optimized roughness and normal maps.* Bottom: *rendering (right) showcasing inconsistent optimizations (red arrows) in a self-similar material (left).*

One possibility is to rely on a tiling strategy. In this context, optimizing for all texels independently i.e., minimizing Eq. (1) for each texel p of the sum, results in noisy output maps and visible seams between adjacent tiles (see Fig. 4, top). Second, materials showcasing self-similarities should be treated consistently by the optimization. However, independently optimizing texels leads to self-similar patches which provide inconsistent results (see Fig. 4, bottom), where similar colored structures tend to result in either smooth or sharp visual transitions.

3.1.2. Material-level learning

To address the aforementioned issues, we propose to train an MLP to achieve spatial regularization (see Fig. 3). This MLP takes as input a small texel patch in the bilinearly upsampled low-resolution SVBRDF $\mathcal{M}_{LR}^{\uparrow\uparrow}$ and outputs a single texel of the high-resolution SVBRDF $\mathcal{M}_{SR}$. This SVBRDF is fed to our differentiable renderer and used to compute the loss $\mathcal{L}$ of Eq. (1), before back propagating through the MLP to evolve its weights. In our setup, each texel p of the high-resolution maps $\mathcal{M}_{SR}(p)$ is computed using the MLP whose entries are a square window of size $N \times N$ centered in p of the input material upsampled bilinearly $\mathcal{M}_{LR}^{\uparrow\uparrow}$. The MLP measures a difference to the center texel, similar to a residual connection and learns a filter that takes as input the raw material upsampled bilinearly and output the upsampled material texel. The resulting model optimizes for a specific pair of shading model $\mathcal{R}$ and radiance upsampler $\mathcal{S}$. Training a neural upsampling filters also allows for tileability preservation, when the input is tileable.

3.2. Metalness regularization

As shown in Fig. 3, base color, normal, metalness and roughness are concurrently optimized. The case of the height map is discussed next. During optimization however, we noticed that the metalness map tends to encode values diverging from its original LR counterpart. To prevent this, we propose a regularization term:

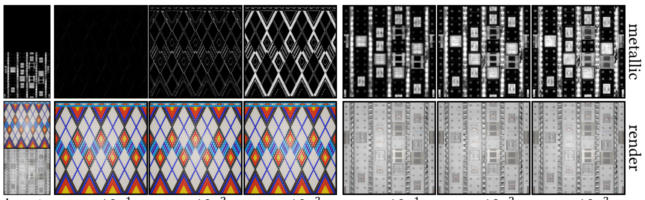


Figure 5: *Comparisons between our inputs and the metalness and render outputs for varying* α*. Our approach prevents value shifts while preserving the aspect of the rendered results.*

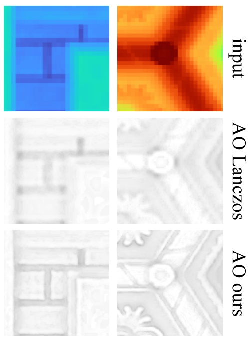
$$\mathcal{L}_{\text{reg}} = \alpha \cdot \sum_{p \in \mathcal{P}} \left(m_{LR}^{\uparrow\uparrow}(p) - m_{SR}(p)\right)^2, \tag{2}$$

where α is a weighting parameter, $m_{LR}^{\uparrow\uparrow}$ denotes a bilinear upsampling of the low-resolution metalness map m_{LR}, and m_{SR} the optimized metalness map. This produces maps more faithful to the original metalness distribution while remaining sharper than the bilinearly upsampled input. We analyze the impact of our regularization term in Fig. 5.

Hence, our final loss takes the form $\mathcal{L} = \mathcal{L}_{\text{fit}} + \mathcal{L}_{\text{reg}}$.

3.3. Height map reconstruction

In the case of materials, most of the geometric mesostructures are captured by the normal map, onto which we focus our efforts, while reconstructing the height map from it afterwards following Martin et al. [MRR*22]. More precisely, we adopt the normal integration scheme proposed in [DC07], which requires a Poisson solve. As noticed by Nehab et al. [NRDR05], geometry reconstruction by normal integration suffers from low-frequency noise. Fortunately, the low-resolution input height map provides us with a dense enough anchor to combat this, by compositing a high-pass filtering of the upsampled normal-based height channel with a bilinear upsampling of the input. Resulting Ambient Occlusion (AO) maps are shown as insets, compared with Lanczos upsampling (the color mapped input LR height is provided on top). We provide more results in the supplementary material.



4. Optimization setup

Data preprocessing As a preprocessing step we render the input low-resolution SVBRDF in the unit square domain under varying point light positions, using an orthographic camera model placed above it. We use 100 single point lights sampled from Fibonacci sequence over the hemisphere [SJP06]. Our differentiable renderer implements a standard microfacet reflectance model [CT82]

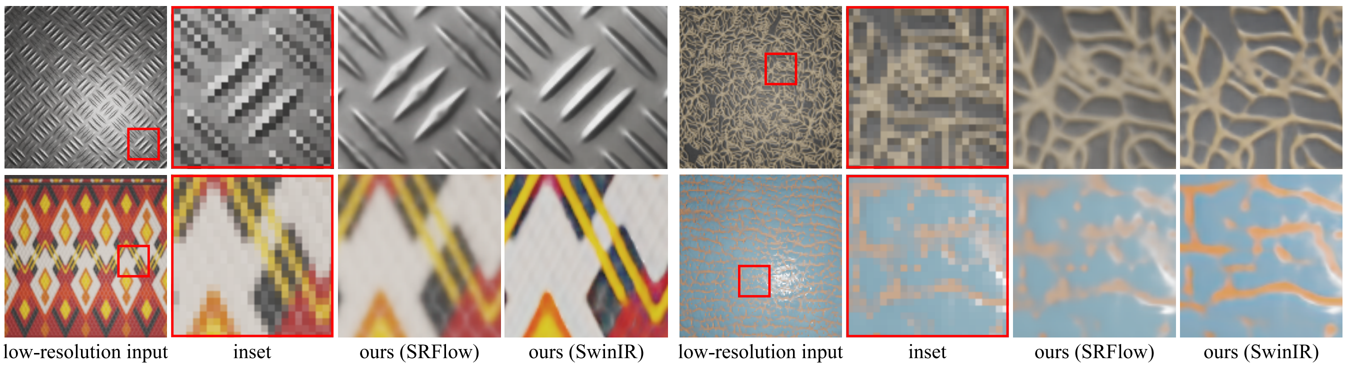


Figure 6: *Comparisons between SRFlow [LDVGT20] and SwinIR [LCS*21] for upsampling SVBRDFs. Since each upsampler comes with its loss and training dataset, the upsampled renderings inherit from the upsamplers' features. SRFlow tends to remain closer in appearance than SwinIR. The latter deviates from the original maps, producing sharper renderings and slight color shifts (best seen in the bottom row).*

with a GGX [WMLT07, TR75] normal distribution function in PyTorch. We then apply an off-the-shelf RGB super-resolution algorithm on those renderings, providing us with pairs of low-resolution SVBRDF and high-resolution renderings with per-pixel correspondence for training our upsampling MLP. As already mentioned, to cope with memory and compute limitations of the state-of-the-art RGB upsamplers we selected, we use crops of size 128^2, which we found sufficient to keep those upsamplers effective while maintaining acceptable performance on a standard machine (upsampling at a rate of 1.6 secs. per batch of 10 images using at most 1.5GB of VRAM).

Optimization details Our MLP consists in 4 fully-connected hidden layers of size 128 with LeakyReLu activations, which takes as input a 17x17 tile of SVBRDF texels (all material channels are concatenated). We use a sliding window mechanism over valid patches to avoid the borders. We opt for $\alpha = 10^{-1}$ in our loss and use a batch size of 128. The networks' weights are optimized using gradient descent with the Adam optimizer with default parameters and a learning rate of 10^{-4}. Differentiable SVBRDF rendering with PyTorch is implemented for GPU execution without CPU control flow or synchronization. Thus, the entire routine can be encapsulated in a single CUDA graph, which minimizes idle time between kernels. Furthermore, we employ CUDA streams to fully overlap the training of the current batch with the copying of the next. This eliminates almost all delays between successive training iterations, which consisted in the slowest part of the pipeline when profiling was carried out. Each epoch takes an average of 3.1 seconds to complete at 512^2 resolution (from 128^2 inputs) on a RTX 3080 Ti, and requires less than 1GB of VRAM. The full method hence takes less than a minute in total for 128^2 inputs (including rendering at low-resolution and upsampling them). We chose to optimize for 10 epochs for all results in the article. Since our method scales linearly with respect to the number of texels, it takes a minute from 128^2 to 512^2, 4 min. from 256^2 to $1K^2$, 16 min. from 512^2 to $2K^2$ and 64min. from $1K^2$ to $4K^2$. Note that these timings consist in the upper bound because it is likely that most materials showcase redundancy which lowers the number of optimization steps required to achieve converged results.

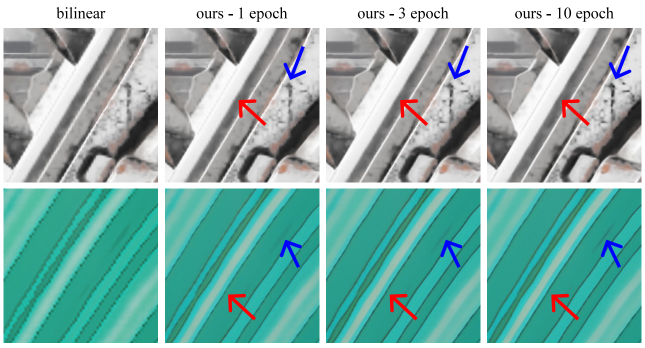


Figure 7: *We show the impact of the number of epochs on the quality of the optimization. Red arrows: the aliasing artifact progressively disappears with more epochs. Blue arrows: Textured elements become more salient with optimization time.*

5. Results and Evaluation

5.1. Results

Choice of super-resolution algorithm We experimented with 4x super-resolution using two efficient learning-based RGB upsamplers: SwinIR [LCS*21] and SRFlow [LDVGT20]. For SwinIR, we opt for the learned weights of the large model *SwinIR-L_x4_GAN* trained for real-world images. For SRFlow, we use the *SRFlow_DF2K_4X* trained weights and a temperature of 0.5. We compare them in Fig. 6 and Fig. 8 (in which we provide additional comparisons to standard filters such as Bilinear and Lanczos).

Progressive training In Fig. 7, we demonstrate the effect of optimizing for multiple epochs. Our neural filter progressively removes aliasing and sharpens the textures of the materials.

$\times16$ **chained upsampling.** We propose to apply chained upsampling using our neural filter twice on the input material. A single level strategy consists in optimizing for x4 upsampling and applying the filter as is to its first output. The two-level strategy reoptimizes a neural filter on upsampled rendering of the first x4 output. As illustrated in Fig. 9, our upsampling neural filter is stable

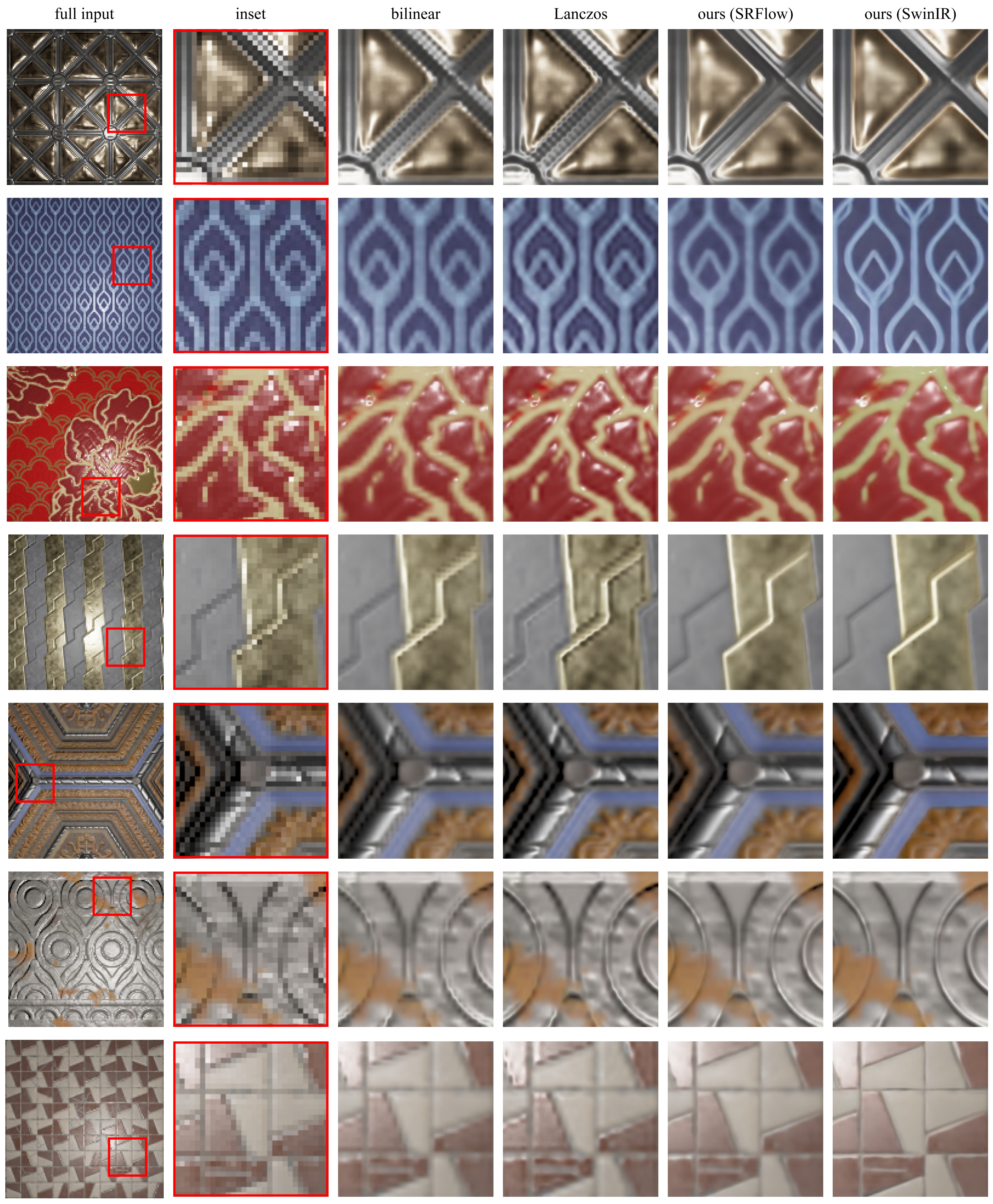


Figure 8: *Results of our method trained using SRFlow [LDVGT20] and SwinIR [LCS*21] upsamplings compared with 4x bilinear and Lanczos upsampling shown in rendered close-ups. Our method provides sharper results in terms of geometry and prevents aliasing compared with the other upsampling methods. Notice how Lanczos filters create overshooting in the color and geometry.*

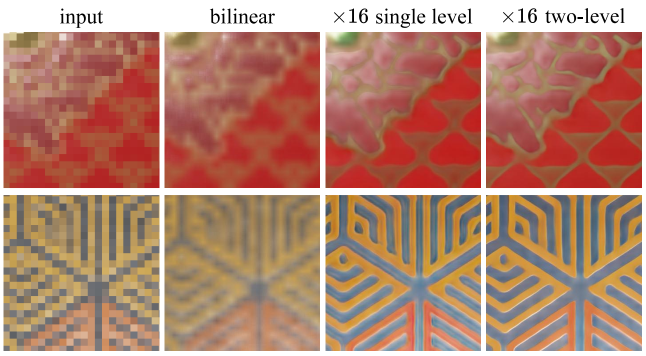


Figure 9: *We achieve x16 super-resolution using SwinIR x4 by applying our neural filter twice to the input. The single level strategy provides usable SVBRDFs but the optimization of the second stage in the two-level strategy brings sharper structures and fewer artifacts.*

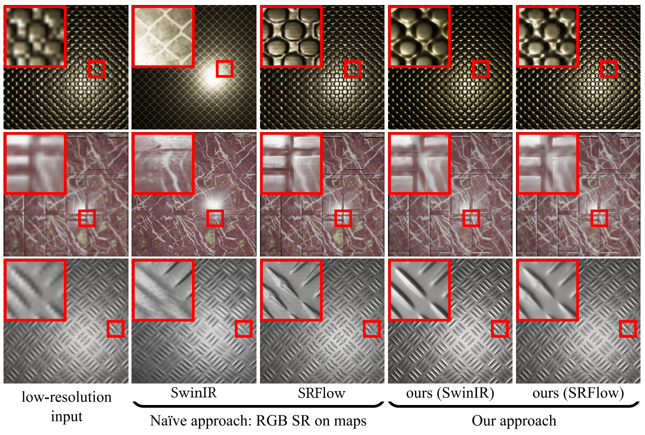


Figure 10: *Direct super-resolution of SVBRDF maps using ML-based RGB upsamplers (2nd and 3rd columns) is not viable, as material maps differ in nature, geometric constraints and statistics from RGB photographs. All maps contribute also in a very different manner to and have a different impact on the final shading.*

enough to be applied successively to its own outputs, but provides sharper results when optimized twice.

We also provide additional results in the supplementary materials. First, we show how our neural upsampling filters can learn on Lanczos-upsampled renderings to provide smoothed yet blurry results and prevent aliasing. Second, we demonstrate how our filters generalize over unseen patches of a material when learned on a partial number of patches. Last, we showcase upsampling on real-world scanned materials (from 4K² to 16K²).

5.2. Comparisons

Basic SVBRDF upsampling We compare MatUp with a baseline consisting in upsampling independently each map using bilinear and Lanczos filters. We compare renderings of these upsampled maps to our method in Fig. 8. These filters only interpolate the

Method	Inputs	Timings
GFT [DDB20]	LR maps + **1 render**	∼ 11-14 min
DIR [GLD*19]	LR maps + N renders	∼ 10 min
MGAN [GSH*20]	LR maps + N renders	∼ 8 min
Ours ($128^2 \rightarrow 512^2$)	LR maps + N renders	**∼ 1 min**
DIR-ref [GLD*19]	**HR maps** + N renders	∼ 1 min

Table 1: *Comparisons with SVBRDF estimation/optimization techniques (for $128^2 \rightarrow 512^2$). DIR-ref: per-pixel refinement step of [GLD*19]. This optimization requires the high-res output.*

signal and lead to staircase effects, aliasing and overshooting (in particular in the normal map) which produces strong visual artifacts at rendering time.

SVBRDF upsampling with ML-trained RGB-upsamplers Using directly RGB upsamplers on SVBRDF maps fails at properly preserving and enhancing the input structures (see Fig. 10). SwinIR fails to handle the heterogeneous content of SVBRDF maps and produces materials which strongly diverge from the inputs (to compare with bilinear upsampling). SRFlow better handles this heterogeneity but systematically adds high-frequency noise to the maps, especially visible in the highlight of specular materials. Overall, processing all maps separately with these ML-based approaches results in uncorrelated signals in the output (Fig. 2–2nd line, Fig. 10).

For the following experiments, we adapt our shading model with our competitors, by extracting diffuse and specular colors from our base color and metallic maps, computed using MatUp and our own shading model. We report each method's inputs and timings in Table 1. The details of the comparison procedure and configurations used are detailed in the supplementary material. Note that the methods to which we compare below do not directly allow for super-resolution. We provide them with our data as inputs to assess their capability to recover higher-resolution SVBRDFs.

Comparison with Guided Fine-Tuning. This approach (GFT) [DDB20] consists in finetuning a SVBRDF estimation network on known SVBRDF patches to later apply the estimation on a single large photograph. Fig. 11 illustrates the results of their method compared with ours. Theirs takes more than 10 mins on a RTX Titan GPU for 256^2 to 1024^2 upscaling, while ours takes less than 2 minutes (5 epochs only at 256²).

Comparison with MaterialGAN. We compare MatUp with the multi-image SVBRDF recovery method of [GSH*20](MGAN) in Fig. 13. Even if the optimization happens in the latent space, the rendering loss is computed in pixel space and optimizing for a full resolution (e.g. 1024^2 or 4096^2), done iteratively using patches of 256^2 texels, which take around 2 minutes for a 64^2 low-resolution input. In comparison, our method takes 2 to 4 minutes to process an input tile of 256^2 to get a 1024^2 output. Our method leads to smoother results, while preserving sharp edges.

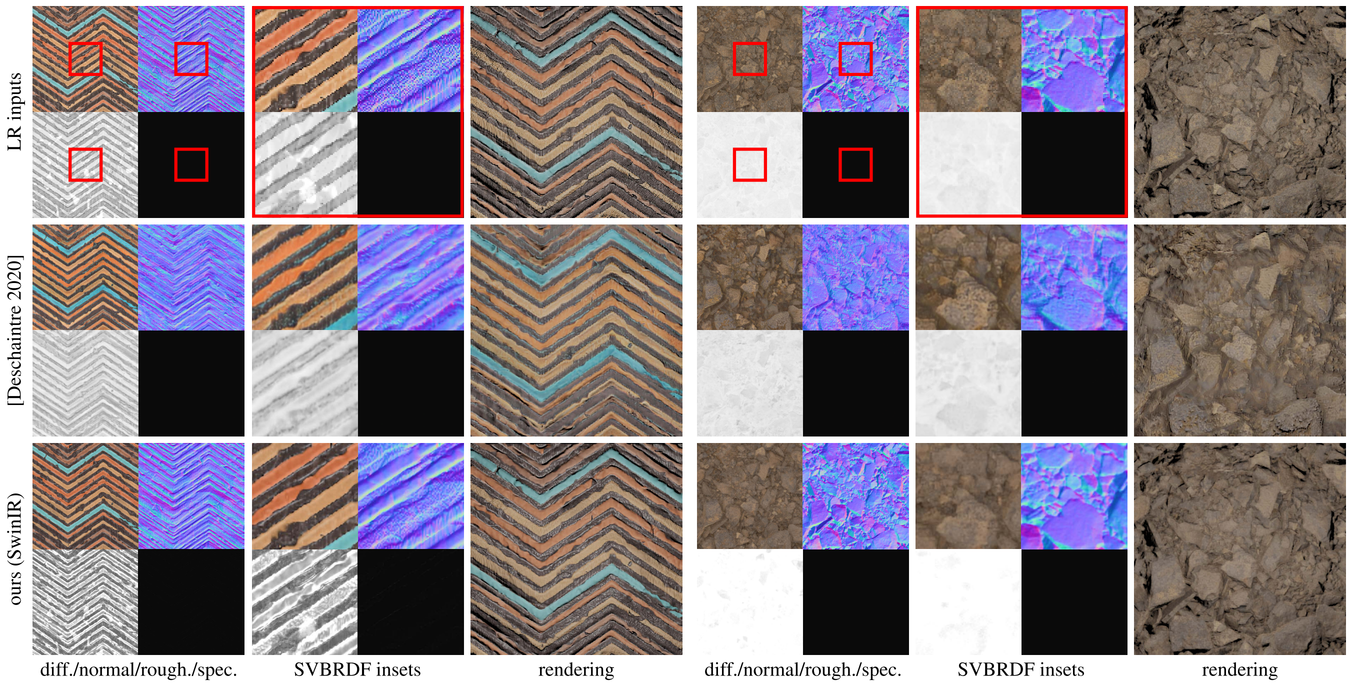


Figure 11: Comparisons with the approach of [DDB20]. *The 1st row shows the LR finetuning guides and corresponding renderings. The estimated maps and renderings are shown in the 2nd row. As expected, the method is unable to provide consistent normals and bakes shading into the SVBRDF because of the lack of information coming from the single input. Our method (3rd row) better preserves the visual aspect of the material when rendered.*

Comparison with Deep Inverse Rendering. We also compare with the method of [GLD*19](DIR) (see Fig. 13, bottom rows), in which we optimize the SVBRDF in the latent space of the learned autoencoder. Their method takes more than 10 minutes for each 512^2 output patch. We showcase the initial inversion in the latent space (*inverse opt.*) and the refinement step (*refinement opt.*), as well as the initial guess provided instead of the SVBRDF estimation used in their paper. Note that the refinement step is exactly the per-pixel approach described in Section 3.1.1, and illustrated Fig. 4.

6. Discussion

6.1. Limitations

Failure cases We present failure cases in Fig. 12. The first row shows a material where the geometry is baked into the base color (see renderings), and the hue is both biased by SwinIR and darkened by both upsamplers. Note that this material has very low roughness, leading to a really sparse signal to process. The second row shows how the artifacts (vertical lines) cause by upsampling using SRFlow with temperature $t = 0.1$ are baked into the SVBRDF. Increasing this value to $t = 0.8$ removes them.

Inconsistent upsamplers When experimenting with recent diffusion-based RGB upsampling methods [RBL*22], we observed blurry artifacts in the upsampled material due to strong inconsistencies in the upsampled renderings. Our data-fitting loss struggles to optimize the SVBRDF when the upsampled renderings contain too few structures persisting across lighting conditions. Temporally-consistent super-resolution of videos has however gained recent interest [CZXL22a, CZXL22b, LFX*22], and we expect that those methods will provide us with additional quality and feature recovery, as material rendering sequences computed

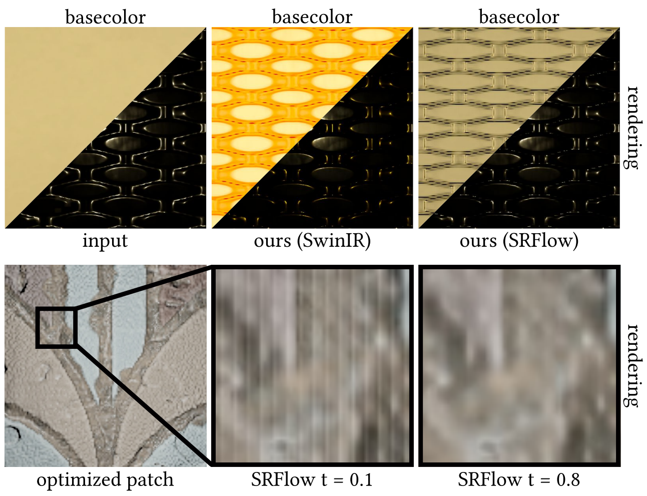


Figure 12: *Failure cases of our approach.*

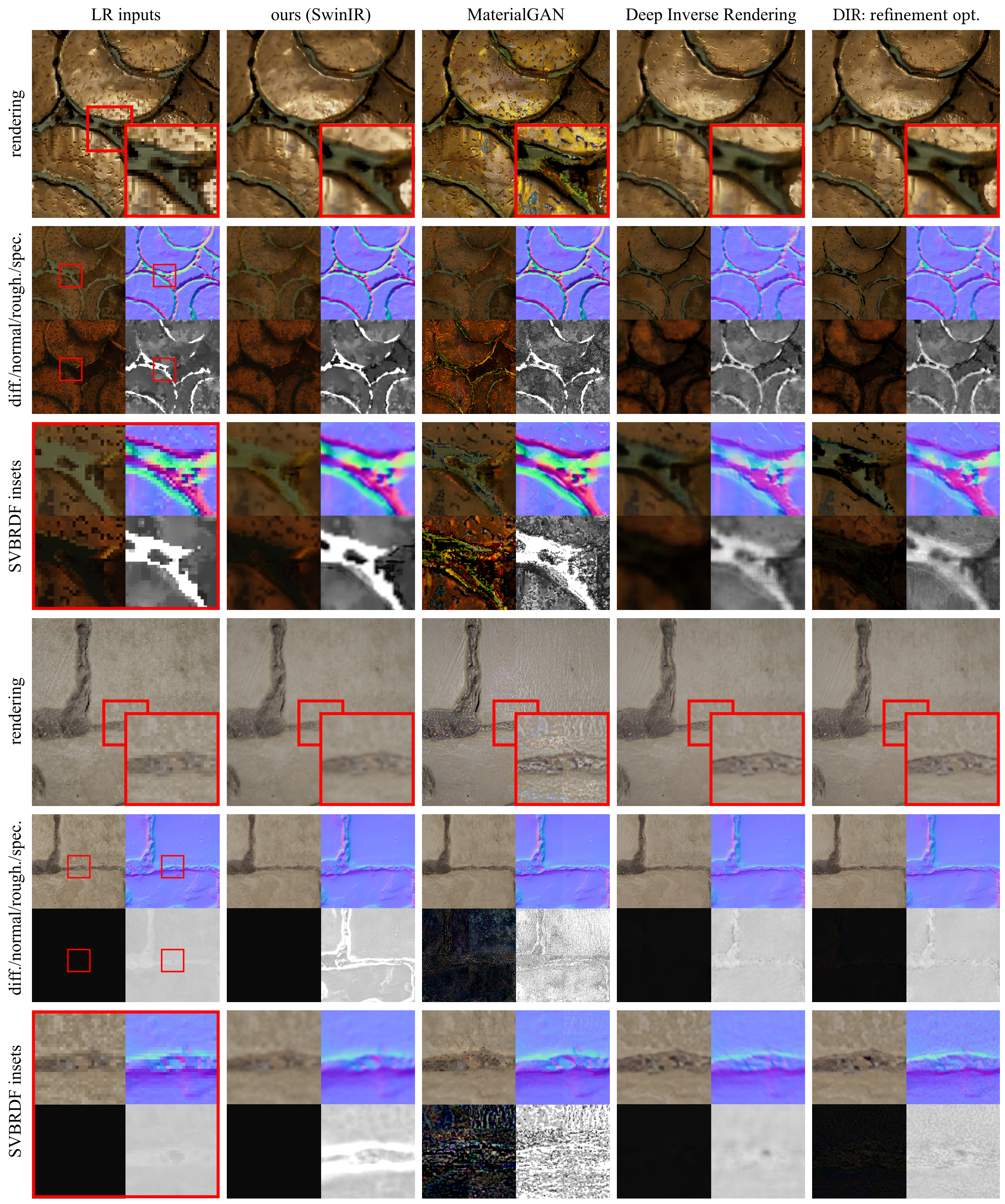


Figure 13: Comparison with MaterialGAN [GSH*20] and Deep Inverse Rendering [GLD*19]. *MaterialGAN (3rd column) exhibits noise, renderings artifacts, and the concatenation of independently upsampled patches leads to visible seems. Deep Inverse Rendering's inversion in the latent space (4th column) leads to biased and blurry SVBRDF, which the refinement step (last column) improves, at the cost of noise. The refinement cannot improve the biased output of the inversion step (see 3rd row), but create sharper results, as shown in both refined outputs materials. Our method (2nd column) leads to smooth and alias-free maps*

with continuously-moving lighting conditions should be an ideal input for those.

Optimizing specular highlights For highly specular materials, recovering signal from point light illumination proves to be feasible since our method has access to the base maps, as opposed to pure SVBRDF estimation methods. Still, our simple point lighting scheme sometimes prevents all regions from being recovered, as shown in the top row of Fig. 12, where the cavities are not well reconstructed. For this task, leveraging other types of illuminant (area light, environment maps, etc.) could help to recover highly specular material information.

Optimization time Although we propose lightweight neural upsampling filters, each optimization relies on a preprocessing step of a dozen of seconds as well as fitting time of less than a minute.

6.2. Future work

Applying the Metappearance framework [FR22] to our pipeline would allow for rapid finetuning, provided the training data. Indeed, experimenting with finetuning the network using about ten samples and handcrafted parameters seems to improve the quality for some of our results. Also, a perceptual loss term for training our network would help to achieve even better detail preservation compared to our sparsity norm, and explore the perception-distortion trade-off for SVBRDFs [BM18]. Finally, experimenting with controllable super-resolution [BM20, RBL*22] could be useful for enhancing artistic control over the upsampled results.

6.3. Conclusion

We proposed MatUp, a lightweight SVBRDF upsampling filter based on radiance super-resolution. We train our neural filter per-material, using a loss defined as a combination of data-fitting and regularization terms, which produces higher resolution SVBRDFs while preserving tileability of the inputs. We demonstrated the superiority of our method compared to a collection of alternative SVBRDF upsampling strategies and justify the need for our learning-based approach by thoroughly comparing with state-of-the art SVBRDF estimation algorithms. MatUp finds application in material capture, for upsampling regions-of-interest, old graphics content update, as well as in generative material upsampling, such methods being currently limited in their native resolution.

7. Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work was mainly conducted during Alban's internship at Adobe Research. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

[AAL16] Aittala M., Aila T., Lehtinen J.: Reflectance modeling by neural texture synthesis. *ACM Trans. Graph. 35*, 4 (jul 2016). 3

[BJMO11] Bach F., Jenatton R., Mairal J., Obozinski G.: Convex optimization with sparsity-inducing norms. 3

[BM18] Blau Y., Michaeli T.: The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 6228–6237. 10

[BM20] Bahat Y., Michaeli T.: Explorable super resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 2716–2725. 2, 10

[BN12] Bruneton E., Neyret F.: A survey of nonlinear prefiltering methods for efficient and accurate surface shading. *IEEE Transactions on Visualization and Computer Graphics 18*, 2 (2012), 242–260. 2

[CHQ*22] Chen H., He X., Qing L., Wu Y., Ren C., Sheriff R. E., Zhu C.: Real-world single image super-resolution: A brief review. *Information Fusion 79* (2022), 124–145. 2

[CT82] Cook R. L., Torrance K. E.: A reflectance model for computer graphics. *ACM Trans. Graph. 1*, 1 (jan 1982), 7–24. 4

[CZXL22a] Chan K. C., Zhou S., Xu X., Loy C. C.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2022). 8

[CZXL22b] Chan K. C., Zhou S., Xu X., Loy C. C.: Realbasicvsr: Investigating tradeoffs in real-world video super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2022). 8

[CZY*19] Cai J., Zeng H., Yong H., Cao Z., Zhang L.: Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (October 2019). 2

[DAD*18] Deschaintre V., Aittala M., Durand F., Drettakis G., Bousseau A.: Single-image svbrdf capture with a rendering-aware deep network. *ACM Trans. Graph. 37*, 4 (jul 2018). 3

[DAD*19] Deschaintre V., Aittala M., Durand F., Drettakis G., Bousseau A.: Flexible svbrdf capture with a multi-image deep network. In *Computer graphics forum* (2019), vol. 38, Wiley Online Library, pp. 1–13. 3

[DC07] Durou J.-D., Courteille F.: Integration of a Normal Field without Boundary Condition. In *Proc. PACV* (2007). 4

[DDB20] Deschaintre 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). 7, 8

[DHI*13] Dupuy J., Heitz E., Iehl J.-C., Poulin P., Neyret F., Ostromoukhov V.: Linear efficient antialiased displacement and reflectance mapping. *ACM Trans. Graph. 32*, 6 (nov 2013). 3

[DRCP14] Di Renzo F., Calabrese C., Pellacini F.: Appim: Linear spaces for image-based appearance editing. *ACM Trans. Graph. 33*, 6 (Nov. 2014). 2

[FR22] Fischer M., Ritschel T.: Metappearance: Meta-learning for visual appearance reproduction. *ACM Trans. Graph. 41*, 6 (nov 2022). 10

[GAD*20] Guehl P., Allègre R., Dischler J.-M., Benes B., Galin E.: Semi-Procedural Textures Using Point Process Texture Basis Functions. *Computer Graphics Forum* (2020). 2

[GFL*22] Gauthier A., Faury R., Levallois J., Thonat T., Thiery J.-M., Boubekeur T.: Mipnet: Neural normal-to-anisotropic-roughness mip mapping. *ACM Trans. Graph. 41*, 6 (nov 2022). 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 (jul 2019). 3, 7, 8, 9

[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). 2, 3, 7, 9

[HDMR21] Henzler P., Deschaintre V., Mitra N. J., Ritschel T.: Generative modelling of brdf textures from flash images. *ACM Trans. Graph. 40*, 6 (dec 2021). 3

[HDR19] Hu Y., Dorsey J., Rushmeier H.: A novel framework for inverse procedural texture modeling. *ACM Trans. Graph. 38*, 6 (nov 2019). 2

[HGH*22] Hu Y., Guerrero P., Hasan M., Rushmeier H., Deschaintre V.: Node graph optimization using differentiable proxies. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. 2

[HHD*22] Hu Y., He C., Deschaintre V., Dorsey J., Rushmeier H.: An inverse procedural modeling pipeline for svbrdf maps. *ACM Trans. Graph. 41*, 2 (jan 2022). 2

[HHG*22] Hu Y., Hašan M., Guerrero P., Rushmeier H., Deschaintre V.: Controlling material appearance by examples. *Computer Graphics Forum 41*, 4 (2022), 117–128. 2

[Kow09] Kowalski M.: Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis 27*, 3 (2009), 303–324. 3

[LCS*21] Liang J., Cao J., Sun G., Zhang K., Van Gool L., Timofte R.: Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 1833–1844. 2, 5, 6

[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 (jul 2017). 3

[LDVGT20] Lugmayr A., Danelljan M., Van Gool L., Timofte R.: Srflow: Learning the super-resolution space with normalizing flow. In *ECCV* (2020). 2, 5, 6

[LFX*22] Liang J., Fan Y., Xiang X., Ranjan R., Ilg E., Green S., Cao J., Zhang K., Timofte R., Van Gool L.: Recurrent video restoration transformer with guided deformable attention. *arXiv preprint arXiv:2206.02146* (2022). 8

[LSM23] Li B., Shi L., Matusik W.: End-to-end procedural material capture with proxy-free mixed-integer optimization. *ACM Trans. Graph. 42*, 4 (jul 2023). 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. 2

[MRR*22] Martin R., Roullier A., Rouffet R., Kaiser A., Boubekeur T.: Materia: Single image high-resolution material capture in the wild. *Computer Graphics Forum 41* (2022). 3, 4

[NRDR05] Nehab D., Rusinkiewicz S., Davis J., Ramamoorthi R.: Efficiently combining positions and normals for precise 3D geometry. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2005) 24*, 3 (Aug. 2005). 4

[OB10] Olano M., Baker D.: Lean mapping. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, Association for Computing Machinery, pp. 181–188. 3

[RBL*22] Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10684–10695. 8, 10

[SHC*22] Saharia C., Ho J., Chan W., Salimans T., Fleet D. J., Norouzi M.: Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 1–14. 2

[SJP06] Swinbank R., James Purser R.: Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography 132*, 619 (2006), 1769–1793. 4

[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 (nov 2020). 2

[TR75] Trowbridge T., Reitz K. P.: Average irregularity representation of a rough surface for ray reflection. *JOSA 65*, 5 (1975), 531–536. 5

[WCH21] Wang Z., Chen J., Hoi S. C. H.: Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 43*, 10 (2021), 3365–3387. 2

[WMLT07] Walter B., Marschner S., Li H., Torrance K.: Microfacet models for refraction through rough surfaces. *Eurographics* (2007), 195–206. 5

[WXDS21] Wang X., Xie L., Dong C., Shan Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 1905–1914. 2

[WYW*19] Wang X., Yu K., Wu S., Gu J., Liu Y., Dong C., Qiao Y., Loy C. C.: Esrgan: Enhanced super-resolution generative adversarial networks. In *Computer Vision – ECCV 2018 Workshops* (2019), Leal-Taixé L., Roth S., (Eds.), pp. 63–79. 2

[YDPG21] Ye W., Dong Y., Peers P., Guo B.: Deep reflectance scanning: Recovering spatially-varying material appearance from a flash-lit video sequence. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 409–427. 3

[YZT*19] Yang W., Zhang X., Tian Y., Wang W., Xue J.-H., Liao Q.: Deep learning for single image super-resolution: A brief review. *Trans. Multi. 21*, 12 (dec 2019), 3106–3121. 2

[ZHD*22] Zhou X., Hasan M., Deschaintre V., Guerrero P., Sunkavalli K., Kalantari N. K.: Tilegen: Tileable, controllable material generation and capture. In *SIGGRAPH Asia 2022 Conference Papers* (New York, NY, USA, 2022), SA '22, Association for Computing Machinery. 2

[ZHD*23] Zhou X., Hasan M., Deschaintre V., Guerrero P., Hold-Geoffroy Y., Sunkavalli K., Kalantari N. K.: Photomat: A material generator learned from single flash photos. In *ACM SIGGRAPH 2023 Conference Proceedings* (New York, NY, USA, 2023), SIGGRAPH '23, Association for Computing Machinery. 2

[ZLVGT21] Zhang K., Liang J., Van Gool L., Timofte R.: Designing a practical degradation model for deep blind image super-resolution. In *IEEE International Conference on Computer Vision* (2021), pp. 4791–4800. 2

[ZSL*17] Zhou Y., Shi H., Lischinski D., Gong M., Kopf J., Huang H.: Analysis and controlled synthesis of inhomogeneous textures. *Computer Graphics Forum 36*, 2 (2017), 199–212. 2