Eurographics Symposium on Rendering - Industry Track (2019)
T. Boubekeur and P. Sen (Editors)

De-lighting a high-resolution picture for material acquisition

Rosalie Martin*, Arthur Meyer*, Davide Pesare.

Allegorithmic, France
*Authors contributed equally

Figure 1: *Left: Input picture in 2048x2048 pixels. Middle: Intermediary result, all 512x512 pixels tiles are de-lit separately and stitched together, seams are visible. Right: Seams have been removed by solving a Poisson system for the full image resolution.*

Abstract

We propose a deep-learning based method for the removal of shades, projected shadows and highlights from a single picture of a quasi-planar surface captured in natural lighting conditions with any kind of camera device. To achieve this, we train an encoder-decoder to process physically based materials, rendered under various lighting conditions, to infer its spatially-varying albedo. Our network processes relatively small image tiles (512x512 pixels) and we propose a solution to handle larger image resolutions by solving a Poisson system across these tiles.

Keywords: De-lighting, Material, Alchemist

1. Introduction

Across industries spanning from fashion to design, from architecture to games, from visual effects to animation, the ability to easily scan materials is becoming more and more desirable. The equipment cost and the expertise needed to acquire materials are quite demanding, thus progress should be achieved to democratize this process. In this context, we focus on the use case of inferring a material from a single image, taken with an arbitrary device, like a mobile phone. In particular in this paper we concentrate on the albedo extraction of this material, which we refer to as *de-lighting*.

Recovering a high quality material from a scan involves accurately capturing its SVBRDF (spatially-varying bidirectional reflectance distribution function), making it reacting to the illumination in a virtual environment as the real material would in the real world. Thus the acquired material must be free of any information coming from the capture environment that is not intrinsic to the material. In the case of the albedo, the resulting map should be free of any illumination effect (shadow, specular highlight, shades), otherwise artifacts would be visible when rendering the material under a new lighting.

Manually removing those effects in the acquired images is a tedious but necessary step. It can be long, difficult and destructive. Mathematical photometry approaches exist that require sequences of pictures of the same sample under varying lighting environment. Unfortunately with a single image the problem is underconstrained. We overcome this issue by leveraging the statistical domain learned by a carefully trained deep convolutional network.

DOI: 10.2312/sr.20191222

Material acquisition requires very high resolution images in order to yield good quality materials. However, in practical applications neural networks can handle limited resolutions due to their high dimensionality. This is why we propose a bucketed approach that processes smaller tiles separately, associated with a post-process to seamlessly merge the de-lit results.

We deployed this approach in a commercial application called Substance Alchemist. Our tool is geared towards productivity, so performance is paramount, which guided some of our final implementation and design choices. However, it targets users without an extensive technical background, photography skills or advanced equipment, so ease of use is also fundamental. Alchemist therefore imposes that we make minimal assumptions on the capturing device, on the quality, lighting and features of the input image and the material subject portrayed in it. A common workflow is the removal of the illumination on outdoor materials, such as forest grounds and stone walls, taken at daylight without constraint on the weather, with an ordinary camera device and without flash lighting.

Thanks to this approach and its performing implementation, we were able to produce high-quality materials that find use in consumer and industrial applications every day.

2. Related Work

Recent works on SVBRDF acquisition from single picture have shown good results using deep learning approaches, often by making assumptions on the material or imposing strong constraints on the material capture.

In their work, Aittala et al. [AAL16] assume that the material is stationary. They also impose the use of the flash light during the material acquisition, as in the recent approaches of Li et al. [LSC18] and Deschaintre et al. [DAD*18]. As those methods are focusing on retrieving the SVBRDF of a material, they are not considering self-projected shadows, and should provide good results on flat materials but may have more difficulties with irregular material geometry. In addition, imposing the use of the flash light during the material capture can be destructive on highly specular materials, as it would produce a saturated area, avoiding a total recovery of the underlying material. Interestingly, all these approaches rely on the U-Net network architecture, introduced by Ronneberger et al. [RFB15]. Indeed it has enabled significant improvements in image-to-image translation tasks by increasing the level of details of the output through the use of skip-connections, that allow fined-grain detail to flow directly to the decoder part while skipping the bottleneck of the network.

Qu et al. [QTH*17] introduce DeshadowNet to perform shadow removal of a single picture. However, in their work the shadows are intrinsically different since they are cast by physical objects, possibly located outside of the picture's field of view, and not by the material itself. Their dataset is made of pairs of pictures with/without the occluder.

Finally, deep learning methods usually operate on small input images, often limited to 256x256 pixels, due to memory and speed limitation, whereas an artist would typically need at least 2048x2048 pixels to produce a valuable material. Perez et al. [PGB03] provides a set of application of the Poisson equation on image processing, that we leverage for this purpose.

3. Method

3.1. Dataset

Although our objective is to remove lighting effects on photographs, creating training pairs using real photographs with/without lighting effects would be almost impossible. Indeed, we could manage taking two pictures of the same material, one with a particular lighting setup to produce shadows, and the other one with a very diffuse lighting setup to avoid shadows and a polarized filter to avoid any specular highlights, however we would always have to deal with shades coming from occluded regions on a highly irregular material. For instance, on a clover ground, regardless of the lighting setting during the capture, we would never get the intrinsic clover color on deep areas. It would also be extremely long to get enough variations in the data and in the lighting conditions to get a representative dataset. Using synthetic data for training a neural network has proven to be efficient and reliable as shown by Li et al. [LSC18], Deschaintre et al. [DAD*18], nonetheless care must be taken to ensure the realism of the synthesized data and their representativeness of the targeted domain.

To construct our dataset, we use the high quality procedural PBR material library Substance Source, that contained around 2000 Substance files at the time of our work. Each Substance file is a node-base graph that outputs the different channels of a physically based spatially varying material (albedo, normal, height, roughness, metallic), with a set of exposed parameters allowing the generation of variations and producing a huge amount of different materials. We decided to focus on outdoor categories (Ground, Stone, Terracotta, Plaster, Concrete-Asphalt) to fit with the most common use cases of Substance Alchemist's users.

We paid a lot of attention in generating realistic materials starting from a Substance file. Indeed, tweaking procedurally the parameters in their allowed ranges could produce highly unrealistic materials and a manual cleaning pass would be necessary, which is time-consuming and discourages from iterating on the dataset. Consequently, we adopt a prudent variation strategy by sampling parameter's variations using a Gaussian distribution centered around the default parameters defined in the Substance file.

Figure 2: *Training pairs. Top: Input, a material rendered under a specific lighting condition. Bottom: Target, the material albedo.*

Once we have satisfactory material variations, we focus on generating lighting condition variations. We use a Substance Designer filter which provides a material renderer, the "PBR Render" node. This filter uses the SVBRDF material information and an HDR environment map to produce a fast and realistic enough rendering, that computes Image Based lighting in addition to casting shadows from the most important light source in the environment map. For this purpose, we have generated a set of minimalist HDR environment maps that emulate an outdoor environment with a sun light at different daytime and a diffuse sky in grayscale, to avoid shifting colors. The corresponding target of each rendered material is the albedo of the material. (See Fig. 2)

We proceed with a final data augmentation step by rotating, flipping, scaling and cropping our images. Starting from roughly 300 Substance files, following this process allows us to first obtain 1800 2048x2048 pixels materials, resulting in 25000 renders and finally around 380000 training pairs of 512x512 pixels images.

3.2. Network architecture

Our network is a deep convolutional network based on the U-Net encoder-decoder, with five levels of convolutional blocks on each side of the latent space, that takes as input a 512x512 lit image and predicts its per-pixel de-lit correspondence. Instead of training the network to output the de-lit image, we train it to compute the illumination map, representing the residual information to add to the input to get the de-lit output, free of shadows and highlights. This improves the quality of the result by reducing the amount of information the network has to output. (See Fig. 3)

We use the L1 norm to compute the loss of the prediction compared to the ground truth. We found out that using Bounded ReLU as activation functions with a threshold of 6 helps the network to converge, by preventing accumulating a perturbation in the input signal across the layers. We use up-sampling with the nearest neighbors in the decoding part, and mirror padding at all stages to reduce the artifacts at the boundaries of each 512x512 tile.

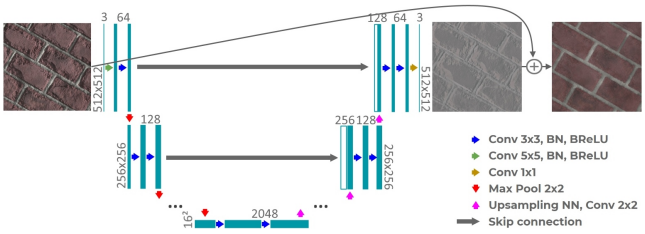

Figure 3: *Network architecture*

3.3. Seamless reconstruction in the gradient domain

To tackle the ability to use this feature on a high-resolution picture, we propose a method to process small data separately, with a merging strategy to produce high-quality results.

We first split the image into 512x512 pixels tiles without overlap, and predict the de-lit version of each tile using our network. We then stitch all the de-lit tiles together, and solve the Poisson equation on each color component separately on the entire image. As an application of Perez et al. [PGB03], we use the color gradients as the guidance field, and provide the entire image borders as the boundary values for the Poisson equation.

This removes the seams visible when all the de-lit tiles are stitched together by smoothing the gradients, and makes the de-lighting feature available on any picture's resolution. This method works particularly well because the tiles are from the same material and present a coherence at their boundaries (See Fig. 1).

4. Results

4.1. Visual quality

Fig. 4 presents some results of inference on synthetic data. Our model successfully removes the shadows and the specular highlights present in the input material, and provides a result that is really close from the ground truth albedo. The details coming from the material elevation are well reduced and the result is still of high quality, without blurry effect.

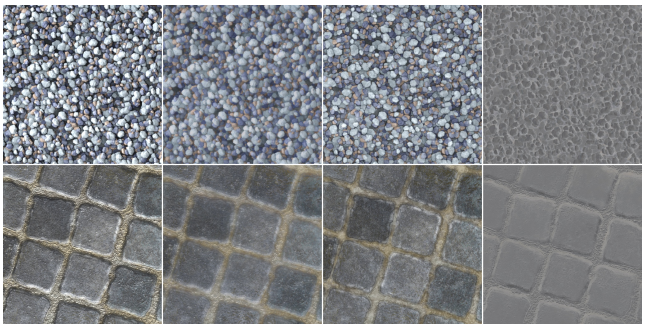

Figure 4: *De-lighting results on the test set. From left to right: Input rendered material (512x512 pixels); De-lit result; Ground truth; Illumination map*

As shown in Fig. 5, our model generalizes well on real pictures, and our merging solution provides a good seamless result for high-resolution input.

Figure 5: *De-lighting results on real use cases. Top: Input picture in 2048x2048 pixels. Bottom: De-lit results after seamless reconstruction*

4.2. Metrics

We evaluate the quality of the delighting using three metrics: Mean Squared Error, Mean Absolute Error and SSIM Error, comparing the de-lit output to the ground truth albedo (See Tab. 1). In addition, we have created a test set dedicated to checking the visual quality of the network output on carefully selected real cases. For this purpose, we have aggregated pictures of grounds and stones taken in the context of photometry and photogrammetry, using several camera devices and under various lighting conditions.

Metric	Lit vs. albedo	De-lit vs. albedo
Mean Absolute Error	0.0864	0.0446
Mean Square Error	0.0120	0.0040
SSIM	0.3023	0.1805

Table 1: *Average metrics computed on the entire test set, comparing the error before and after the de-lighting.*

4.3. Implementation details and performance

We use TensorFlow framework in Python to train our model. Our network requires around 20Go of memory during the training stage. We use a NVivia Quadro GV100 for the training, and it takes around five days to get a fully converged model. For the deployment into Substance Alchemist software, we first used TensorFlow C++ API, built for GPU running, but we figured out that the model loading and inference time was unstable across GPUs. For instance the first inference was 7 to 30 times longer than the next inferences, which made the feature unusable.

To overcome this issue we have implemented a C++ API for our network directly using CUDA and CuDNN libraries. The weights of the network are exported as a binary file that is read by the C++ API when the network is rebuilt, and we have developed some optimized CUDA kernels for the operations that are not natively available in CuDNN (for instance, the reflect padding and the nearest neighbor up-sampling). We did a benchmark on the performance, to compare our CuDNN implementation with the use of TensorFlow GPU C++, and also the use of single or half precision in CUDA, as well as the use of the optimized TensorCore operations for the convolution with CuDNN (See Tab. 2).

Method	Duration (ms)
TensorFlow 1st inference	306.7
TensorFlow next inference	44.3
CuDNN FP32	53.1
CuDNN FP16	37.0
CuDNN TensorCore optim	21.1

Table 2: *Inference performance benchmark. Mesure done on a NVidia Quadro GV100 for a batch of 16 images in 512x512, expressed in ms per image.*

The Poisson equation solving is done using the Intel®MKL Poisson Library, using the gradients as vector field. The gradients in horizontal axis and vertical axis are computed per tile and per channel (RGB) in a dedicated CUDA kernel, then stitched together as for the de-lit tiles. We use the Dirichlet boundary condition and fix the boundary values to the actual values of the entire stitched image boundaries. We set a 0 value of gradient at the boundary of a tile in the considered direction (e.g. we have a null gradient for all pixels of the right border in the horizontal gradient, and respectively for the bottom border and vertical gradient).

Finally, de-lighting a 2048x2048 picture (a batch of 16 512x512 tiles) takes around 1 second, the first half for the de-lighting task and the second half for the seams removal.

5. Discussion

We have proposed a method for removing all the illumination visible on a picture of a material in order to retrieve the albedo of the captured material. Our solution shows good results at solving this under-constrained problem thanks to the domain learning using a deep convolutional network. We have developed a framework for the procedural data generation starting from Substance files, and produced a huge and realistic dataset of synthetized rendered materials.

Our solution shows limitation if the shadows are too strong though, or if they are projected too far from the occluding part. In these cases, it fails at recovering entirely the albedo but still provides an improvement compared to the input.

In future work, we plan to extend this solution to extract other channels of the SVBRDF of the captured material (normal, height and roughness). We also plan to train the model to remove shadows that are projected by objects in addition to the self-occlusion, to cover more use cases of Substance Alchemist's users.

6. Acknowledgements

We thank Romain Rouffet for providing real case pictures coming from his work on photometry. We also thank Jerome Derel and Tamy Boubekeur for the fruitful discussions at the early stage of the development, and Jeremy Levallois and Maxime Morel for their help in the deployment of the de-lighter into Substance Alchemist. Arthur Meyer is now working at Coupang, South Korea.

References

[AAL16] AITTALA M., AILA T., LEHTINEN J.: Reflectance modeling by neural texture synthesis. *ACM Transactions on Graphics 35*, 4 (2016), 1–13. 2

[DAD*18] DESCHAINTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Single-image SVBRDF capture with a rendering-aware deep network. *ACM Transactions on Graphics 37*, 128 (2018), 15. 2

[LSC18] LI Z., SUNKAVALLI K., CHANDRAKER M.: Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. *Lecture Notes in Computer Science 11207 LNCS* (2018), 74–90. 2

[PGB03] PEREZ P., GANGNET M., BLAKE A.: Poisson Image Editing. *ACM Transactions on Graphics* (2003), 313–318. 2, 3

[QTH*17] QU L., TIAN J., HE S., TANG Y., LAU R. W.: DeshadowNet: A multi-context embedding deep network for shadow removal. *Proceedings - 30th IEEE Conference on CVPR*, 1 (2017), 2308–2316. 2

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science 9351* (2015), 234–241. 2