Author's Version to appear in Computers & Graphics

Deep scene-scale material estimation from multi-view indoor captures (supplementary)

Siddhant Prakash[a], Gilles Rainer[a], Adrien Bousseau[a], George Drettakis[a]

[a] *Université Côte d'Azur and Inria, Sophia-Antipolis, 06902, France*

ARTICLE INFO

Article history:
Received 18 July 2022
Accepted 28 September 2022
Available online 10 October 2022

Keywords: Material estimation, Deep learning, Indoor scenes, Photogrammetry, Synthetic dataset, Digital 3D assets

ABSTRACT

The movie and video game industries have adopted photogrammetry as a way to create digital 3D assets from multiple photographs of a real-world scene. But photogrammetry algorithms typically output an RGB texture atlas of the scene that only serves as visual guidance for skilled artists to create material maps suitable for physically-based rendering. We present a learning-based approach that automatically produces digital assets ready for physically-based rendering, by estimating approximate material maps from multi-view captures of indoor scenes that are used with retopologized geometry. We base our approach on a material estimation Convolutional Neural Network (CNN) that we execute on each input image. We leverage the view-dependent visual cues provided by the multiple observations of the scene by gathering, for each pixel of a given image, the color of the corresponding point in other images. This image-space CNN provides us with an ensemble of predictions, which we merge in texture space as the last step of our approach. Our results demonstrate that the recovered assets can be directly used for physically-based rendering and editing of real indoor scenes from any viewpoint and novel lighting. Our method generates approximate material maps in a fraction of time compared to the closest previous solutions.

Author's Version Preprint.

1. Network Architecture Details

We provide network details for the diffuse track and specular track in Tables 1 and 2 respectively for an image of resolution $640\times384\times3$. As described in Section 4.2 in the main paper, the number of input channel is defined by the multi-view statistics being fed to each network. In the decoder from layer 4 onwards, a nearest-neighbor upsampling is followed by concatenation of encoder features, and two convolutions. For activations, each convolution/de-convolution layer is followed by a leaky ReLU with a weight 0.2 for the negative part.

Input	Layer	Output
In: $640 \times 384 \times 9$	4×4 Conv, 64, stride 2	**1**: $320 \times 192 \times 64$
0: 9	FC + SeLU	**1**: 128
1: $320 \times 192 \times 64$	4×4 Conv, 128, stride 2	**2**: $160 \times 96 \times 128$
1: 256	FC + SeLU	**2**: 256
2: $160 \times 96 \times 128$	4×4 Conv, 256, stride 2	**3**: $80 \times 48 \times 256$
2: 512	FC + SeLU	**3**: 512
3: $80 \times 48 \times 256$	4×4 Conv, 512, stride 2	**4**: $40 \times 24 \times 512$
3: 1024	FC + SeLU	**4**: 512
4: $40 \times 24 \times 512$	4×4 DeConv, 256, stride 1	**5**: $80 \times 48 \times 256$
4: 768	FC + SeLU	**5**: 256
5: $80 \times 48 \times 512$	4×4 DeConv, 128, stride 1	**6**: $160 \times 96 \times 128$
5: 384	FC + SeLU	**6**: 180
6: $160 \times 96 \times 256$	4×4 DeConv, 64, stride 1	**7**: $320 \times 192 \times 64$
6: 192	FC + SeLU	**7**: 64
7: $320 \times 192 \times 128$	4×4 DeConv, 3, stride 1	**Out**: $640 \times 384 \times 3$
7: 67	FC + SeLU	**8**: 3

Table 1: Details of the network architecture for diffuse track.

e-mail: siddhant.prakash@inria.fr (Siddhant Prakash), gilles.rainer@inria.fr (Gilles Rainer), adrien.bousseau@inria.fr (Adrien Bousseau), George.Drettakis@inria.fr (George Drettakis)

Input	Layer	Output
In: $640 \times 384 \times 18$	4×4 Conv, 64, stride 2	**1**: $320 \times 192 \times 64$
0: 9	FC + SeLU	**1**: 128
1: $320 \times 192 \times 64$	4×4 Conv, 128, stride 2	**2**: $160 \times 96 \times 128$
1: 256	FC + SeLU	**2**: 256
2: $160 \times 96 \times 128$	4×4 Conv, 256, stride 2	**3**: $80 \times 48 \times 256$
2: 512	FC + SeLU	**3**: 512
3: $80 \times 48 \times 256$	4×4 Conv, 512, stride 2	**4**: $40 \times 24 \times 512$
3: 1024	FC + SeLU	**4**: 512
4: $40 \times 24 \times 512$	4×4 DeConv, 256, stride 1	**5**: $80 \times 48 \times 256$
4: 768	FC + SeLU	**5**: 256
5: $80 \times 48 \times 512$	4×4 DeConv, 128, stride 1	**6**: $160 \times 96 \times 128$
5: 384	FC + SeLU	**6**: 180
6: $160 \times 96 \times 256$	4×4 DeConv, 64, stride 1	**7**: $320 \times 192 \times 64$
6: 192	FC + SeLU	**7**: 64
7: $320 \times 192 \times 128$	4×4 DeConv, 4, stride 1	**Out**: $640 \times 384 \times 4$
7: 67	FC + SeLU	**8**: 4

Table 2: Details of the network architecture for specular track.

2. Additional Results

We provide additional results for our experiments.

2.1. Ablation

Figure 1 shows the effects of increasing multi-view information for the SYNTHETIC VEACH AJAR scene. This figure is a extension of Fig. 12 in the main paper.

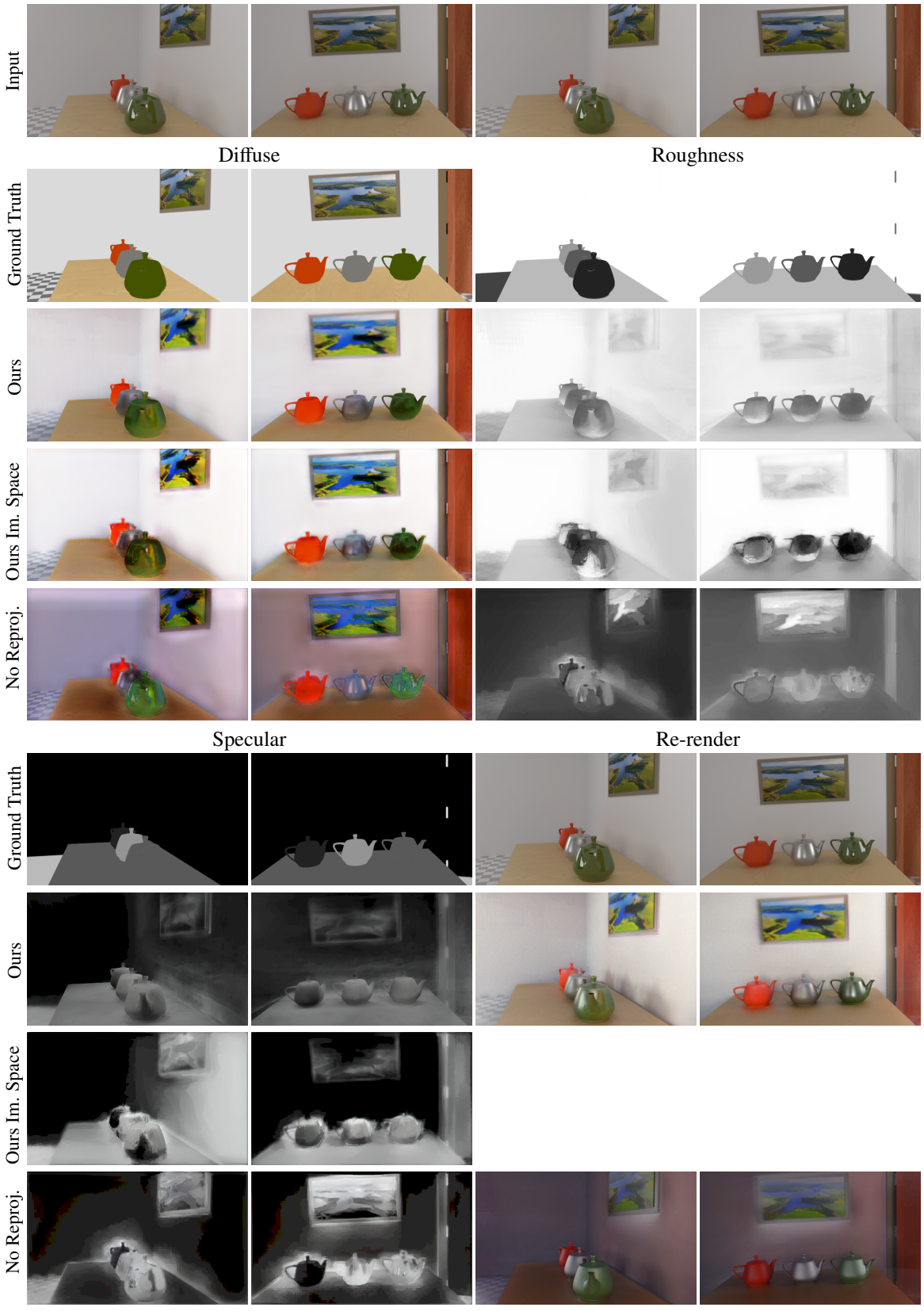

Figure 1: Example images from Synthetic Veach Ajar showing the effect of increasing multi-view information on results. Note how the quality of the maps is significantly improved by using the reprojected statistics observed in our image space predicted maps as compared to no reprojection, i.e. using only a single image. Furthermore, gathering the image space maps in texture space helps improve the consistency of the maps across views and thus improves re-rendering by assigning same material in local regions (esp. in roughness and specular maps). As a result, we get better re-rendering as the re-rendered images are closer to ground truth.