Eurographics Symposium on Rendering (DL-only Track) (2021)
A. Bousseau and M. McGuire (Editors)

Supplemental: Material and lighting reconstruction for complex indoor scenes with texture-space differentiable rendering

Merlin Nimier-David[1,2], Zhao Dong[1], Wenzel Jakob[2] and Anton Kaplanyan[1]

[1]Facebook Reality Labs, USA
[2]EPFL, Switzerland

Figure 1: *Our method enables photorealistic relighting of captured scenes. Here, we simulate lighting variations at three different times of day (**a-c**) in the* ROOM-0 *scene. The scene reacts correctly to new illumination, which differs significantly from the original indoor lighting it was captured with (**d**).*

In this document, we provide additional analyses and results. Specifically, we include new results in Section 1, a description of the dataset in Section 2, studies of the sensitivity of our pipeline to corrupted inputs in Section 4, an additional comparison to previous work in Section 3, detailed pseudocode in Section 5, and finally a validation of the method on synthetic reference data in Section 6. Please see the **supplementary video** for an animated version of these results.

1. Additional results

As shown in the main paper, the scenes reconstructed by our method are readily used in photorealistic applications such as relighting for dataset augmentation. Figure 1 shows a daylight simulation setup in the ROOM-0 scene. To achieve this, we lit the scene with an HDR environment map after removing its windows (as delimited by the instance segmentation) and "turning off" existing emitters (i.e. simply setting all emission to zero). Despite being captured in strictly local indoor illumination (Figure 1, **d**), the reconstructed scene reacts faithfully to natural external illumination.

We also show side-by-side comparisons between the re-rendered scenes and the actual camera captures on additional scenes in Figure 9. The view-dependent effects, such as glossy highlights, were correctly disentangled and re-simulated using a physically based renderer, which allows the method to closely match the reference photos.

Finally, Figure 2 visualizes our optimized high-resolution albedo textures in unwrapped UV space. Inspecting the scene's recovered albedo confirms that our method correctly disentangled lighting and reflectance, as there are no visible residual shadows or glossy reflections visible in the texture.

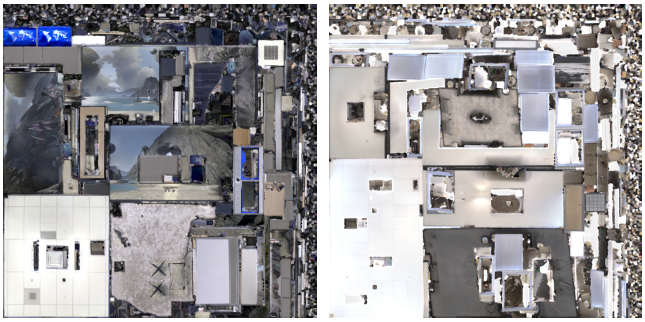


Figure 2: *Our method uses texture space sampling to uniformly sample the scene's parameters during optimization. We visualize the optimized albedo texture of scenes* OFFICE-0 *and* ROOM-0 *in texture space (i.e. unwrapped UV space). The 4K textures were downsampled before embedding into the document due to size constraints.*

2. The Replica dataset

Replica [SWM*19] is a recently published high-quality dataset comprising of large interior scenes captured in high resolution using several sensors. It includes geometry (a single large mesh per scene), instance labels. The capture was performed by an operator walking through the scene, holding a specialized rig.

One of the sensors is an RGB camera that captures a video stream in Standard Dynamic Range (SDR). Extended dynamic range is achieved by cycling exposure time every frame, as shown in Figure 3. Therefore, there is no High Dynamic Range data available for any given specific viewpoint. As with any real-life capture, video frames must be corrected for camera calibration: color shift, vignette, and distortions are removed in a preprocessing step.

The dataset also includes a fused HDR texture for each scene. However, unlike our method's results, that texture has all shadows, view-dependent effects and global illumination "baked-in" and is therefore not suitable for relighting, scene editing, realistic re-rendering with correct view-dependent effects, etc.

3. Comparison to Li *et al.* [LSR*20]

The method of Li *et al.* [LSR*20] produces an impressive decomposition (albedo, normals, depth and local lighting) from *a single input image*. However, it is important to note that these predictions are only valid for the input viewpoint. This is suitable for object insertion and material editing applications, which only need to composite the effect *over the original image*. In contrast, our method requires more inputs (multiple views, geometry) but can fully re-render the scene from any viewpoint with high fidelity, which enables VR applications and full relighting.

Nevertheless, we run author-published code on one of the input frames of scene OFFICE-2 for a qualitative comparison, shown in Figure 4. Understandably, re-rendering the scene using the predicted quantities (albedo, local illumination, etc) does not capture all high-frequency detail, surface emitters and view-dependent effects.

	Original	**Displacement noise** (σ)			**Decimated**		
		0.5 cm	1.0 cm	3.0 cm	75%	50%	10%
MSRE	**0.124** ±0.074	0.125 ±0.069	0.127 ±0.063	0.163 ±0.062	0.125 ±0.075	0.127 ±0.077	0.128 ±0.076

Table 1: *The reconstruction quality, evaluated via re-rendering error on 30 viewpoints selected at random in the* OFFICE-0 *scene, is robust to perturbations of the input geometry. The corresponding corrupted geometry is illustrated in Figure 6.*

4. Sensitivity to input quality

In order to test the robustness of our pipeline, we study its performance when provided lower quality or corrupted inputs.

Instance segmentation We first study the influence of the quality of the surface segmentation. Our method relies on such a segmentation to limit the spatial variations of roughness and specular parameters to a plausible extent, as well as to guide emitter detection.

The Replica dataset [SWM*19] used in this paper provides a good (although slightly coarse) instance segmentation. In this experiment, we automatically create an alternative segmentation: we first attribute separate UV islands to separate classes, and additionally split classes where there is high curvature. This operation is complete in a few seconds and we did not tweak the heuristics per scene.

Note this does not result in a valid or meaningful instance segmentation, but does achieve our goal of roughly separating objects in order to reduce the degrees of freedom for emission, roughness and specular parameters. Running our method using the low-quality automatic segmentation rather than the higher quality segmentation provided by the Replica dataset results in a reconstructions of equivalent quality, as shown in Figure 5.

Input geometry Next, we evaluate the impact of geometry quality on the reconstruction, using scene OFFICE-0 as an example. We use two types of perturbations, visualized in Figure 6: the geometry is either corrupted by normally-distributed displacement noise or severe decimation. As for the ablation study (Section 4.3 of the main text), we re-render the scene after reconstruction from 30 viewpoints chosen at random and compute the pixelwise MSRE. The quantitative results are given in Table 1. We find that unless the corruption is unrealistically large, the reconstruction copes well with

Figure 3: *Scenes from the Replica dataset [SWM*19] were captured with a handheld camera rig. The RGB video stream uses multiplexed HDR: SDR frames alternate between three exposure times (*$1/100\,\mathrm{s}$*, *$1/1666\,\mathrm{s}$* and *$1/16666\,\mathrm{s}$*). For each camera pose, we therefore have access to a single SDR frame only.*

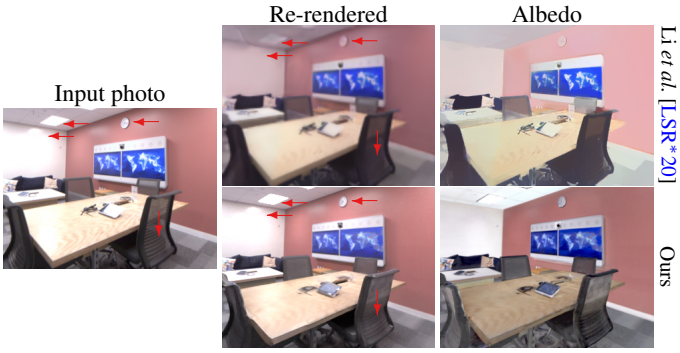


Figure 4: *Comparison to the method of Li* et al. *[LSR*20] on scene* OFFICE-2. *Since it does not target applications requiring re-rendering the scene from arbitrary viewpoints, the output of [LSR*20] understandably lacks some high-frequency detail, surface emitters and glossy reflections.*

the altered geometry. In particular, even significant decimation only slightly impacts re-rendering quality.

Capture conditions We study the impact of lighting conditions at capture time on the quality of the reconstruction. Keeping camera positions and all other parameters fixed, we create three synthetic datasets from the scene of Section 6. The wall and ceiling emitters' radiance values are set to $(15, 15)$, $(0, 20)$ and $(25, 3)$ respectively.

Since the ground truth values are known, we directly compute the L_2 difference in each case, shown in Figure 7. The optimization generally converges well. In the third configuration, we found that the darkest shadowed area converged slower (at equal iteration count). This could be improved using e.g. a relative loss function.

5. Texture-space optimization algorithm

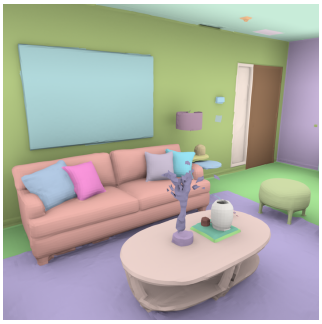
The pseudocode for one step of our texture-space optimization scheme is given in Listing 1. It constitutes the inner loop of our algorithm, and is executed repeatedly until a fixed number of iterations is reached or convergence is achieved. Note that this algorithm is not specific to our implementation choices, and could be directly adapted to any differentiable renderer, loss function and optimizer.

6. Validation on synthetic data

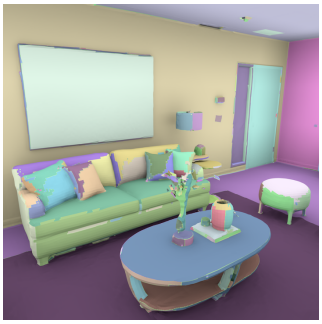
We validate that our method can correctly disentangle physically based illumination and material parameters on a controlled synthetic scene, where the exact ground truth values are known. Reference images are computed by rendering multiple views with path tracing until convergence. These converged images along with the segmented scene geometry are then given as input to our method.

The optimization's results are shown in Figure 8. Emission, albedo and roughness properties are recovered perfectly. Our method detects light sources and their intensity, while the emission of objects that do not emit light is correctly set to zero. High frequency variations of the scene's albedo are accurately reproduced in the optimized 8192×8192 texture. Note this represents over *200 million parameters*.

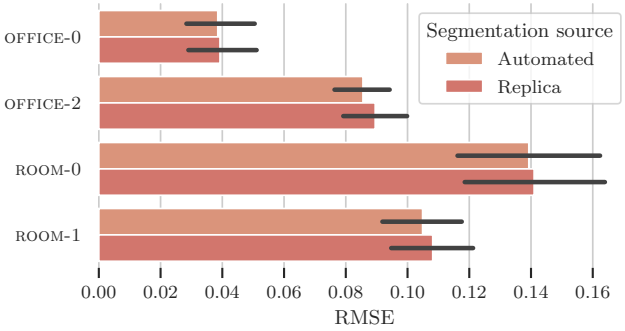
Two diffuse objects were assigned different specular values than



(a) *Instance segmentation provided with the Replica dataset [SWM*19].*

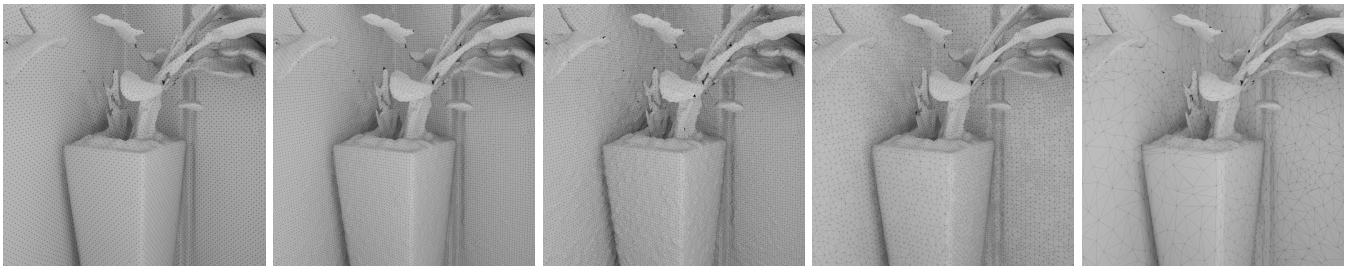


(b) *Naive surface segmentation auto-generated from surface curvature and the automatic UV mapping.*



(c) *Our method performs equality well with either type of surface segmentation, as measured by re-rendering error on a set of 30 views chosen at random.*

Figure 5: *Studying the sensitivity of the reconstruction to the quality of the surface segmentation used.*



(a) *Original geometry* **(b)** *Displaced* ($\sigma = 0.5\,\text{cm}$) **(c)** *Displaced* ($\sigma = 3.0\,\text{cm}$) **(d)** *Decimated (50% left)* **(e)** *Decimated (10% left)*

Figure 6: *Sensitivity to input geometry quality. We corrupt the input geometry with either normally-distributed displacements or severe decimation (the triangulation is visible when zooming-in). The corresponding reconstruction errors are shown in Table 1.*

Emitters' radiance	Base color	Emission	Roughness	Specular
$e_1 = 15, e_2 = 15$	0.1183	0.0003	0.0308	0.0859
$e_1 = 0, \quad e_2 = 20$	0.1147	0.0002	0.0342	0.1312
$e_1 = 25, e_2 = 3$	0.1338	0.0002	0.0648	0.1184

Figure 7: *Sensitivity to capture conditions. Reconstructing a synthetic scene in three lighting conditions (bottom) globally converges well. We found that the dark, heavily shadowed area converged slower (top).*

the ground truth (door panel and table legs). This is due to a remaining ambiguity: when an object is not glossy (i.e. it has medium to high roughness), a given brightness can be explained equally well with a slightly higher albedo or a higher specular value. We do not address it explicitly in our method as the end effect is minor: both results are plausible and achieve the correct brightness and appearance.

References

[LSR*20] LI Z., SHAFIEI M., RAMAMOORTHI R., SUNKAVALLI K., CHANDRAKER M.: Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proc. of CVPR* (2020), IEEE. 2, 3

[SWM*19] STRAUB J., WHELAN T., MA L., CHEN Y., WIJMANS E., GREEN S., ENGEL J. J., MUR-ARTAL R., REN C., VERMA S., CLARKSON A., YAN M., BUDGE B., YAN Y., PAN X., YON J., ZOU Y., LEON K., CARTER N., BRIALES J., GILLINGHAM T., MUEGGLER E., PESQUEIRA L., SAVVA M., BATRA D., STRASDAT H. M., NARDI R. D., GOESELE M., LOVEGROVE S., NEWCOMBE R.: The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* (2019). 2, 3, 4, 7

```python
def optimization_step(scene, optimizer, inv_uv, spp):
  uv_sub = . . .     # Select a subset of UV space
  # Lookup the corresponding 3D mesh position for each texel using the inverse UV mapping.
  mesh_positions = lookup_mesh_positions(inv_uv, uv_sub)
  ref_views = . . . # Select a batch of random reference camera positions.
  # Check visibility from the camera positions.
  positions = cull_with_frustum(mesh_positions, ref_view)
  positions = check_visibility(positions, ref_view)
  # Connect `spp` primary rays from the camera to the selected mesh positions.
  # Also lookup reference RGB values corresponding to each ray.
  rays, ref_values = create_primary_rays(ref_views, positions, spp)
  # Evaluate each pixel color with path tracing.
  values = differentiable_path_tracing(scene, rays)
  loss = loss_function(values, ref_values) # E.g. L2 loss
  # Backpropagate through the rendering algorithm.
  gradients = backpropagate(loss)
  # Gradients account for observations from all directions included in the batch.
  optimizer.step(gradients)
```

Listing 1: *Pseudocode for one iteration of our texture-space optimization scheme.*

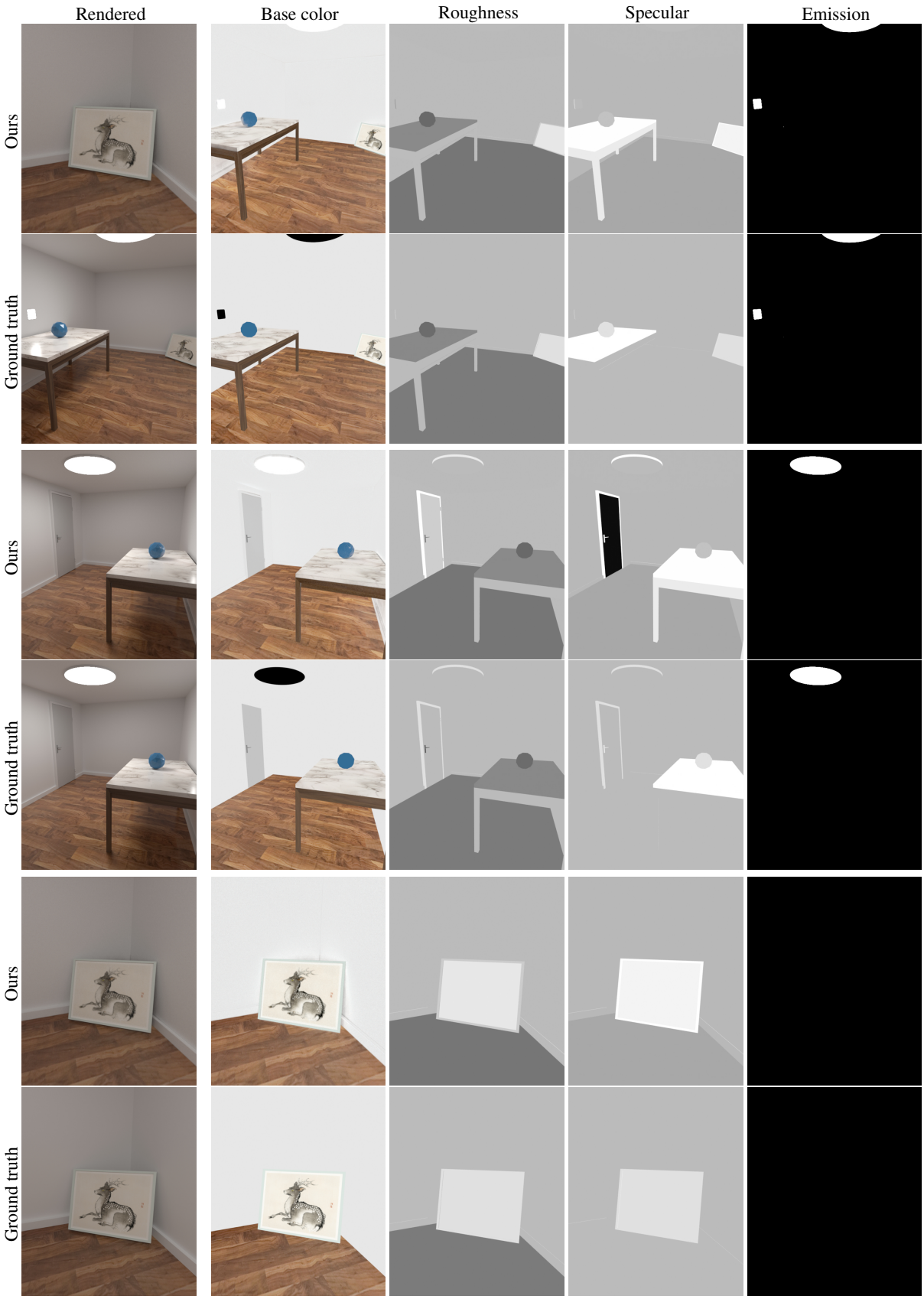


Figure 8: *Validation on a synthetic scene. In this controlled setting, we can compare our method's results to known ground truth values. Scene parameters are correctly recovered and disentangled.*

Figure 9: *Additional results on four scenes of the Replica dataset [SWM*19]. We compare re-rendered scenes reconstructed with our method against reference frames from the same viewpoint.*