Supplemental Material for Shape from Tracing: Towards Reconstructing 3D Object Geometry and SVBRDF Material from Images via Differentiable Path Tracing

1. Hyperparameters

To reconstruct each view, our differentiable path tracer renders an image from the appropriate viewpoint with 64 samples per pixel using 2 ray bounces, beginning at a resolution of 128×128 pixels. Surprisingly, we found that this low pixel count provides enough spatial resolution to reconstruct detailed surface geometry and material appearance, given a sufficient number of views. For our mesh colors materials, we use a texture resolution level of $r = 3$, which we found to be sufficient given our geometry subdivision and simplification steps.

To perform gradient-based optimization, we use the Adam optimizer [5]. In our experiments, geometry optimization requires more fine-tuning than material optimization, thus we use a step size of 10^{-2} for texture optimization and 10^{-4} for geometry optimization. We also tune the momentum hyperparameters for each of these stages:we set the momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.99$ during material optimization, and use their default recommended values during geometry optimization.

All results were produced on desktops with an AMD Ryzen 2700X and a GTX 1080Ti using our variant of the GPU version of Redner.

2. SVBRDF Comparison

Methodology To isolate and evaluate the material reconstruction capabilities of differentiable pathtracing, we compare with the work of Deschaintre et al. [4]. Here, we perform material optimization in a coarse-to-fine manner on a planar surface perpendicular to an orthographic camera; optimization starts at a resolution of 4×4 and increases to 256×256 in power of 2 increments. Lights positions are sampled randomly from the surface hemisphere.

As the two renderers have different implementations of the Torrance-Sparrow material model, we compare image differences to the respective ground truth renderings instead of comparing each method's renders directly to each other. Upon this, we evaluate MSE, PSNR, and SSIM on re-renders as well as for normals and each of the individual material maps.

Data We use the test set provided by the single-view version [3] of the comparison work which consists of 38 spatially-varying BRDFs. Each material map is bounded to the range $[10^{-4}, 1.0]$ to account for differences in treatment of low roughness between material model implementations.

Results Coarse-to-fine texture optimization produces comparable error in final re-renderings (black line, Fig. 1 right; qualitative results to left), with decreasing error in material map reconstruction error as the number of input views increases. We can also see that our method optimizes an incorrect 'perfect' re-rendering reconstruction with just one view, as SVBRDF reconstruction does not have a unique solution. This is expected given that our choice of optimization loss function employs no prior, while the comparison condition has a strong data-driven one.

There are trade offs between these two methods. Whereas optimization through differentiable pathtracing can produce better re-renderings and at arbitrary resolution, the priors of Deschaintre et al. produce more plausible material maps and allows faster test time. A natural question is how to have the best of both worlds: high render accuracy, speed, and well-behaved material maps. Theoretically, training a neural network with a differentiable renderer should yield a good initialization which could then be improved by direct optimization using the same renderer. We leave such an experiment for future work.

3. Shadow Art

The ability to optimize geometry to match the output of a physically-based renderer has other applications other than straightforward 3D reconstruction. One such application is shadow art: producing geometry which casts particular shadows when illuminated from specific angles. Prior work has demonstrated computational tools for solving this problem, using either geometric optimization [6] or stochastic search [7]. The geometry optimization step of our system naturally supports shadow art production with only small modifications: one simply changes the target images to those of the desired shadows. We construct a scene with two

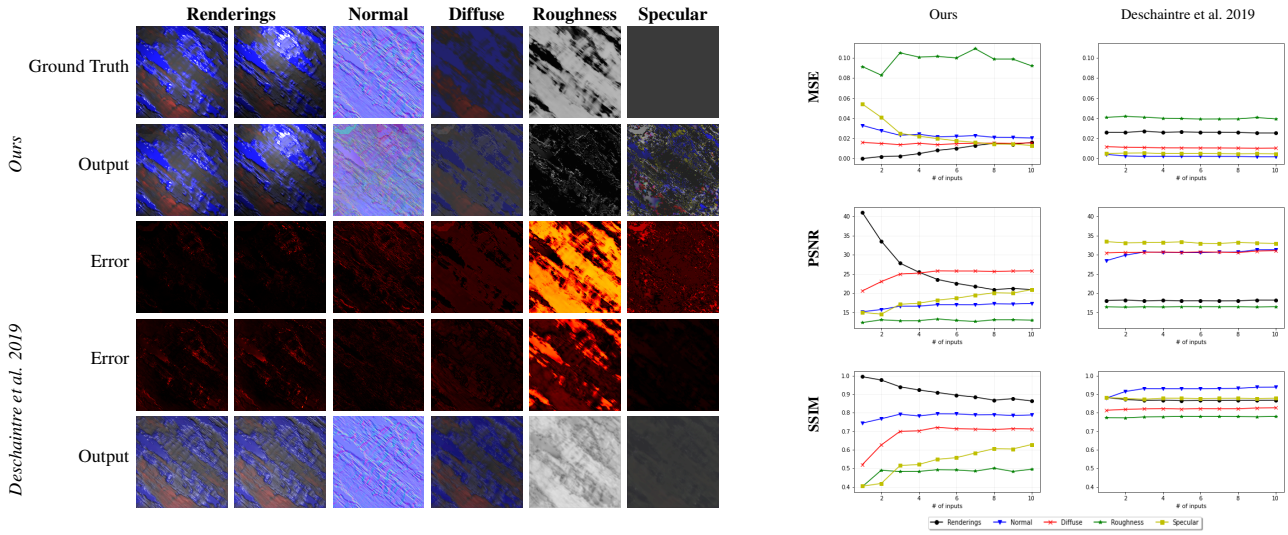

Figure 1. *Left:* SVBRDF reconstruction of a spatially varying wood-like texture given 10 input views. Diffuse and specular albedo maps have a gamma correction of 2.2 applied. Note the comparatively high variation of our material maps despite better re-rendering error. This is due to the lack of a prior and the non-unique optimization landscape–particularly in the intertwined roughness and specular maps. *Right:* Reconstruction metrics plotted as a function of number of input views. Our optimization benefits more from an increased number of views. Note also the stabilization in output quality around 8–10 views. This occurs when it is no longer able to "cheat" in the optimization space and perfectly capture particular view points.

shadow receivers: diffuse white planes orthogonal to each other, forming walls like the corner of a room. Starting from a sphere mesh with a radius of 0.25, we optimize the geometry so that it projects a desired shadows onto each wall when lit from behind. As before, we use an image reconstruction loss, in this application comparing the target shadow and rendered shadow an the end of each iteration. Because the cameras are positioned to only see the projection of the 3D object onto 2D planes, the ground truth is ambiguous and the mesh can converge to any one of many solutions, most of which exhibit degraded geometry. In order to preserve the quality of the mesh, we remove the self-intersecting geometry by applying the surface resampling and remeshing step described in section 3.3 more frequently than we would with more informative targets. Figure 2 shows an example.

4. Environment Illumination

We also demonstrate that our approach can reconstruct under environment map lighting, too. Setup: Stanford dragon under Grace Cathedral [2] illumination. Figure 3 shows results and compares to point illumination. More general environment map lighting comes with slight degradation of both material and geometry reconstruction.

5. Improving Neural Reconstruction

We use Pixel2Mesh++, a recent deep neural network for converting multi-view imagery into a mesh [8], to reconstruct a couch from 3D-R2N2's dataset of rendered ShapeNet models. One issue with this deep reconstruction network is that increasing the number of input views does not significantly increase reconstruction quality, and so the number of available input views is limited to three for training efficiency. As such, we use Wen et al.'s three-view network which, according to the authors, performs closely to models trained with more views. Then, we use this output as initialization for our 32-view refinement (Fig. 4). While we expect our refinement to capture more detail than the network's coarse approximation because we can exploit the increased number of input views, we draw attention to how our pipeline recovers from incorrect initial geometry. In this way, our pipeline is a convincing post-process to learned network priors.

6. Capture Setup

Figure 5 shows the capture setup we used to gather input for our real-world reconstruction experiment. We use a Pixel 3A phone to capture video by walking around the target object, and we capture an environment map by fusing together multiple exposure brackets taken using an Insta360 ONE 360 camera.

7. Effect of Ambient Occlusion

Here we investigate the effect of self-shadowing and ambient occlusion on geometry and diffuse material reconstruction. We set up a virtual scene in which our camera is focused on the surface of a sphere covered in small

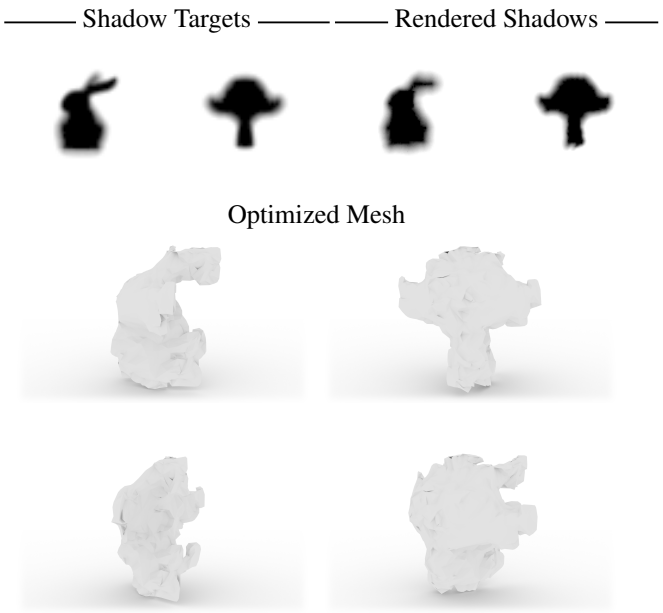

Figure 2. Shadow art example. The top row shows two target shadows provided to our pipeline cast by meshes of a bunny and a monkey, as well as the shadows cast by the mesh that we produce cast onto two orthogonal white walls. The bottom two rows show the output mesh at multiple views. Due to the regular resampling steps, the mesh maintains a relatively smooth surface despite the sparse target information. Note that because our pipeline produces physically-correct shadows, our targets can contain gradients of soft shadows. In contrast, prior work on computational shadow art assumed binary mask images as shadow targets. Finally, because we optimize through gradient descent and are constrained by only two views, the output mesh converges to a different shape every time , each of which can cast shadows that approximate the targets.

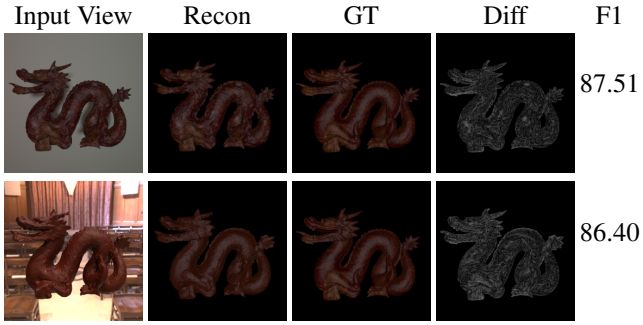

Figure 3. Reconstruction under different illumination models; point light *(Top)* and environment map lighting *(Bottom)*. To account for the extra cameras used for multiple light views in the former case (32 cameras with two different light positions), the number of cameras was doubled in the latter (64). Difference maps use a 2x multiplier. F1 uses a tolerance of 0.01.

bumps. In this case, we compare three strategies: full global illumination, local illumination with path-traced shadows, and purely local illumination. Figure 8 shows the results. All versions are able to successfully reconstruct the geometry, though introducing full global illumination does improve the result. Both the local and path-traced shadows methods erroneously darken the surface albedo in valleys that have self-shadowing. With full global illumination, the optimizer disambiguate interreflection from surface albedo and recover better estimates of the texture and the geometry.

Figure 4. Our refinement using a geometry initialization from inputting 3 views from the 3D-R2N2 dataset [1] *(Left)* into Pixel2Mesh++ [8]. The deep reconstruction *(Middle)* is quite coarse. We can upsample this initialization using our refinement pipeline recover much more of the object's detail, such as its legs, cushion seams, and sharp edges *(Right)*. Training a neural network on a large number of views is quite inefficient and, if instead using these two methods together, largely unnecessary for retrieving the utmost reconstruction detail. Even the coarse reconstruction from a 3-view deep network is a sufficient geometry initialization for our refinement. We report the F1 score of both geometry reconstructions with a tolerance of 0.05; our pipeline has significant improvement over its initialization.

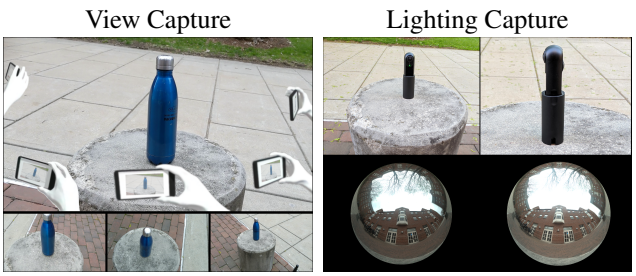

Figure 5. Our capture setup. (a) video capture of an object outdoors with smartphone, (b) acquisition of HDR light probe with consumer 360 camera.

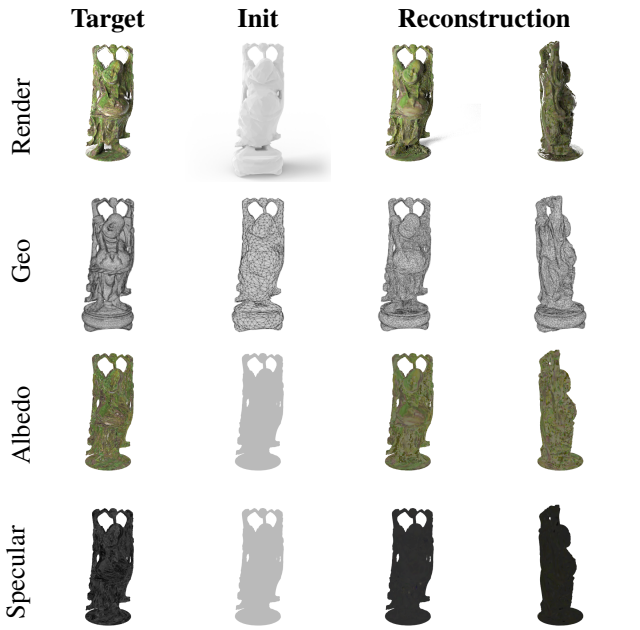

Figure 6. Reconstruction of the Buddha model with a nature texture, from a target scene where it was inside the Cornell Box. The optimization used 32 cameras distributed in a Fibonacci sphere, each with 2 light angles, making for a total of 64 images. The image resolution was 128×128. Column 2 shows initializations; column 4 shows reconstructions under a novel view. Fine geometric details on the model's head and robes are captured in our reconstruction, as well as details in the albedo. The target specular texture exhibits some variation; at the expense of some parts of this detail, our specular reconstruction matches the artistic preference of a relatively uniform specular roughness.

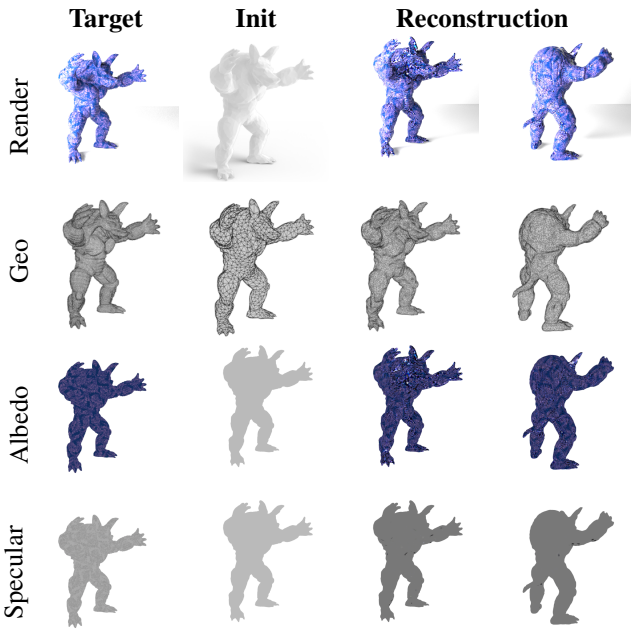

Figure 7. Summary of our reconstruction of the Armadillo model inside the Cornell Box, with a blue pattern. The optimization used 32 cameras distributed in a Fibonacci sphere, each with 2 light angles, making for a total of 64 images. The image resolution was 128×128. Column 2 shows initializations; column 4 shows reconstructions under a novel view

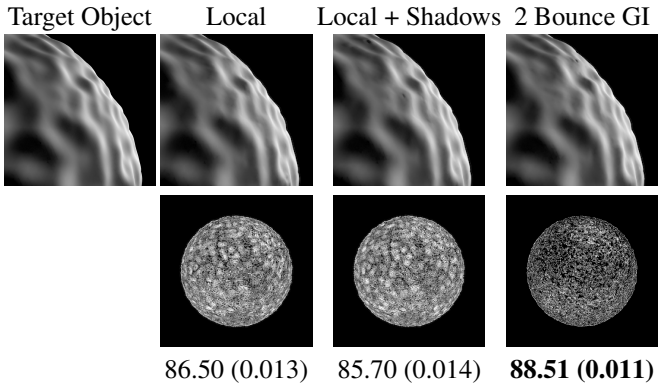

Figure 8. A bumpy, displacement-mapped sphere *(Left)* reconstructed using local illumination *(Top Left)*, local illumination with path-traced shadows *(Top Middle)*, and full global illumination *(Top Right)*. Middle row shows the difference (intensity magnified $25\times$) of each reconstruction's albedo with the uniform gray target albedo. The numbers in the bottom row are "F1 Score (Albedo mean absolute error)." Introducing soft path-traced shadows and global illumination not only improves the optimizer's ability to reconstruct the geometry of the bumps, but also prevents these lighting effects from being baked into the surface's albedo.

8. Additional Simulated Results

Figures 6 and 7 show additional results from reconstructing simulated objects.

References

[1] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3

[2] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 189–198, 1998. 2

[3] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, 37(128):15, aug 2018. 1

[4] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau. Flexible svbrdf capture with a multi-image deep network. *Computer Graphics Forum(Eurographics Symposium on Rendering Conference Proceedings)*, 38(4):13, jul

2019. 1

[5] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR 2015*, 2015. 1

[6] N. J. Mitra and M. Pauly. Shadow art. *ACM Transactions on Graphics*, 28(5), 2009. 1

[7] D. Ritchie, B. Mildenhall, N. D. Goodman, and P. Hanrahan. Controlling Procedural Modeling Programs with Stochastically-Ordered Sequential Monte Carlo. In *SIGGRAPH*, 2015. 1

[8] C. Wen, Y. Zhang, Z. Li, and Y. Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation, 2019. 2, 3