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

Purvi Goel[1, 2] Loudon Cohen[1] James Guesman[1] Vikas Thamizharasan[1]
James Tompkin[1] Daniel Ritchie[1]

[1]Brown University [2]Stanford University

Abstract

Reconstructing object geometry and material from multiple views typically requires optimization. Differentiable path tracing is an appealing framework as it can reproduce complex appearance effects. However, it is difficult to use due to high computational cost. In this paper, we explore how to use differentiable ray tracing to refine an initial coarse mesh and per-mesh-facet material representation. In simulation, we find that it is possible to reconstruct fine geometric and material detail from low resolution input views, allowing high-quality reconstructions in a few hours despite the expense of path tracing. The reconstructions successfully disambiguate shading, shadow, and global illumination effects such as diffuse interreflection from material properties. We demonstrate the impact of different geometry initializations, including space carving, multi-view stereo, and 3D neural networks. Finally, with input captured using smartphone video and a consumer 360° *camera for lighting estimation, we also show how to refine initial reconstructions of real-world objects in unconstrained environments.*

1. Introduction

Reconstructing digital representations of the appearance of objects is important to many industries, including visualization, cultural heritage, and entertainment. At a minimum, this task requires estimating the shape of the object via its surface geometry, and estimating the material appearance properties of the object. Recreating these properties accurately by hand requires skill and labor, so automatic reconstruction techniques are useful to complete this task.

Many techniques have been proposed with a common high-level approach: capture multiple views of the object with an imaging sensor, often under varying illumination, to describe the underlying geometry and material properties under appearance assumptions. These techniques can be forward or 'bottom up,' by directly estimating object properties from observed sensor data, or can be inverse or 'top down,' by optimizing an underlying model until its rendering is consistent with the captured sensor data.

For bottom-up methods, multi-view stereo approaches directly estimate the depth of points on the object surface from calibrated RGB cameras, under a Lambertian surface reflectance assumption. Time-of-flight and structured light sensors can also directly estimate depth under simplified reflectance assumptions; depth point clouds can then be fused into volumes for surface reconstruction. Photometric stereo approaches use RGB cameras to directly estimate surface normal directions from objects exposed to light from different directions, typically with non-spatially-varying surface albedo and Lambertian or restricted BRDF reflectance models. These material reflectance assumptions cause limitations or inaccuracy in complex shape and material reconstruction. Further, methods may also be limited by their light transport assumptions, e.g., that no diffuse interreflection exists for Lambertian materials.

Top-down approaches suffer these problems in reverse, as the renderer must be able to accurately reproduce the appearance of objects under as few assumptions as possible for shape, material, and light transport. While realistic rendering is possible, any renderer must also be efficient to use in optimization to fit a model to the captured camera view. That is, it must provide gradients which describe the direction of error with respect to the object's shape and material. As such, many differentiable renderers support only simplified camera and geometry (e.g., simplified visibility [26, 34]), simplified material (e.g., diffuse only [14]), or simplified light transport (e.g., rasterization [41]).

However, differentiable *path tracing* methods [18, 30] capable of simulating global illumination can reproduce and optimize complex appearance with fewer assumptions about geometry, material, and light transport. Path tracing is a theoretically elegant approach, but its application to multi-view object reconstruction is difficult in practice due to the computational complexity of computing derivatives with respect to the object's shape and material properties.

In this paper, we investigate how to reconstruct an object from multi-view images via differentiable path tracing. Given multiple calibrated views of an object under known lighting, represented either by point lights or an HDR environment map, we explore how to reconstruct both the 3D geometry of the object as a surface mesh and the surface material as a spatially-varying Torrance-Sparrow BRDF model. We refine an initial coarse mesh, produced by any one of a variety of reconstruction methods, at the triangle level with a mesh colors SVBRDF representation. This combination provides coarse-to-fine optimization of both shape and material through geometry subdivision, simplification, and remeshing stages.

We discover with simulated objects that this approach can reconstruct fine geometric and material detail from low-resolution (128×128) target camera views. These reconstructions include the disambiguation of shading and shadows from material variation, the disambiguation of global illumination effects on surface albedo like color bleeding from diffuse interreflection, and the reconstruction of spatially-varying materials with different roughness and specularity. Our efficient representations provide reconstructions within a few hours of optimization, versus naive approaches which can settle at incorrect local minima during gradient-based optimization for inverse rendering.

In addition, we use physically-based differentiable path-tracing to reconstruct from nearly-unconstrained unstructured real-world data. Given only a hand-held smartphone camera video of a target object and an environment map captured by a consumer HDR 360 camera, we explore the challenging problem of reconstruction 'in the wild.'

In short, we show that efficient representation and optimization of surface geometry and material makes differentiable path tracing a promising technique for high-quality object reconstruction. We contribute:

- An investigation into benefits, limitations, and design choices (e.g., parameter space, optimization ordering, and initialization choices) for applying differentiable path tracing to joint geometry/SVBRDF reconstruction in both simulated and real-world settings.

- A differentiable mesh colors texture representation [48] suitable for optimization problems involving meshes with continually-evolving topology.

Our code and real-world data is available at http://www.github.com/brownvc/shapefromtracing. This includes our implementation of mesh colors [48, 27], which to our knowledge has no prior public implementation.

2. Related Work

We focus our discussion on differentiable rendering as applied to inverse problems and on methods which recover shape and spatially-varying non-diffuse material.

2.1. Differentiable rendering

With aims to optimize through or "invert" the rendering process, the past decade has seen many efforts to develop renderers which are differentiable in output pixels with respect to different input scene properties [26, 34, 22, 3, 10]. Modern deep learning toolkits such as Tensorflow and Pytorch3D also now provide differentiable rendering, currently through rasterization [41, 33]. These renderers operate on mesh representations of 3D geometry; parallel efforts have also explored differentiable variants of ray marching for rendering implicit surfaces [12, 23, 25, 38, 28]. All of the above consider either only geometry, or geometry plus local illumination. Recently, differentiable formulations of global illumination rendering have been proposed, resulting in physically-based inverse renderers [18, 30, 49].

Differentiable renderers have been used to fit morphable human face models to images [7, 4] and to optimize for more general classes of objects [47, 22, 3, 32], to acquire materials [19] and optimize for effects like caustic reflections [30], paired with an encoder to predict subsurface scattering parameters [2] and to simultaneously estimate materials and lighting in 3D scenes [1]. We show that geometry and material refinement via differentiable physically-based rendering can account for complex light transport effects. This strategy also makes it feasible to reconstruct real-world objects exhibiting reflections, specular highlights, and soft shadows, within unconstrained environments, given calibrated views and an HDR environment map.

2.2. Geometry and material reconstruction

Many works reconstruct geometry and material; we refer to Weinmann et al. [42] for a recent review. Geometry methods include multi-view stereo [37, 11] techniques to reconstruct point clouds with diffuse color, space carving [16] techniques to reconstruct voxel volumes with diffuse color, or photometric stereo techniques to reconstruct surface normals [43, 17] and spatially-varying specular materials [8].

Some approaches reconstruct complex material with simplified geometry. Lin et al. [21] present a shape-agnostic method for on-site BRDF capture, and Gao et al. [5] use data-driven methods to reconstruct SVBRDFs under planar assumptions. Other methods implicitly perform reconstruction via view synthesis. Xu et al. [46] use data-driven photometric stereo to generate new views from sparse views, which then drive reconstruction via multi-view stereo [35]. Li et al. [20] present a learning-based method to reconstruct SVBRDF and geometry from a single image.

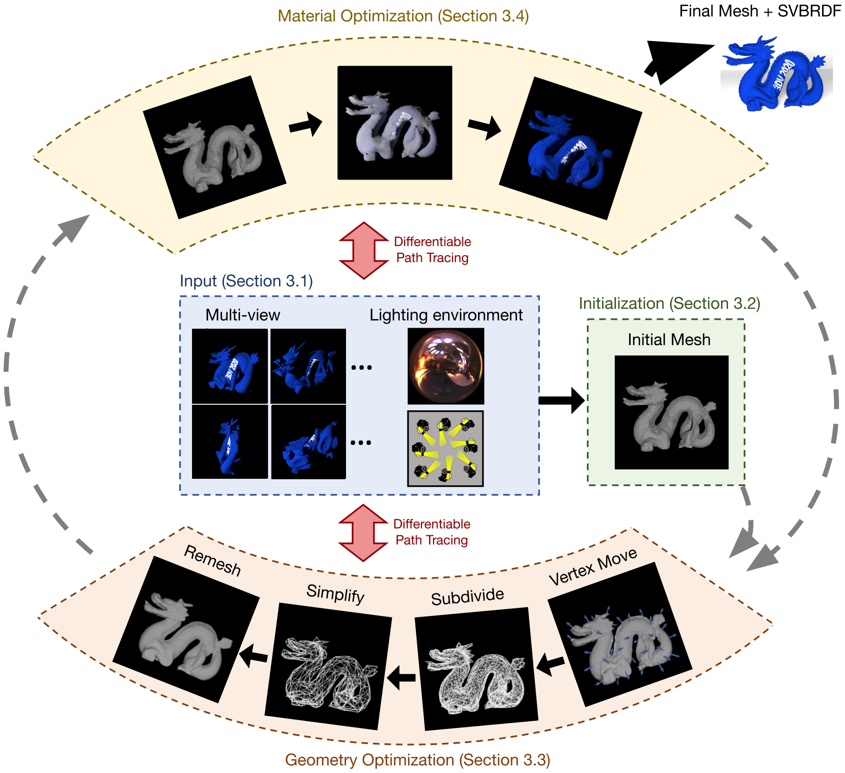
Material Optimization (Section 3.4)
Final Mesh + SVBRDF
Differentiable Path Tracing
Input (Section 3.1)
Multi-view
Lighting environment
Initialization (Section 3.2)
Initial Mesh
Differentiable Path Tracing
Remesh
Simplify
Subdivide
Vertex Move
Geometry Optimization (Section 3.3)

Figure 1. Proposed steps for simultaneous geometry and spatially varying material reconstruction via inverse path tracing. Given input views of an object under known lighting, our pipeline begins with a coarse shape initialization, produced through any one of a number of approaches such as space carving or multiview stereo. It then alternates between material and geometry optimization, adjusting surface mesh vertices and SVBRDF material texels via stochastic gradient descent with respect to path traced renderings of the current reconstruction. This process is made coarse-to-fine by the subdivision of the surface geometry, which then implicitly and automatically subdivides the surface texture via a multi-scale per-facet SVBRDF texture representation.

Other methods reconstruct spatially-varying BRDFs with specular components and whole-object geometry. Tunwattanapong et al. [40] use a dense lighting capture setup and turntable to simulate varying spherical harmonic environment maps. Xia et al. [45] reconstruct geometry and SVBRDF under unknown illumination from coarse initializations by using temporal traces of the reflected illumination as the object rotates over time, though it cannot handle interreflections or occlusions. Kang et al. [13] use a controlled light box with matching synthetic training data to learn detailed geometry and SVBRDF reconstruction, though it cannot handle interreflection and self-shadowing. Most flexibly, Nam et al. [29] present a practical smartphone-based geometry and SVBRDF capture system which uses interactive inverse-rendering, although the system is constrained to blacked-out room with point illumination. None of these approaches explicitly model global illumination effects like interreflection and self-shadowing.

Some methods explicitly model interreflection. Lombardi and Nishino [24] model multiple bounces of light through path tracing and compute derivatives with respect to reflectance and illumination. Geometry adjustment is modeled from an initial depth fusion through a linear combination of surface normals, which can inflate or deflate the surface. Park et al. [31] model interreflection and Fresnel reflectance in their learning-based recovery of scene properties from RGBD imagery, via surface light field and specular reflectance map reconstructions. Both approaches assume accurate geometry initialization, while we include results on reconstructions from coarser initializations. Overall, the problem of simultaneous geometry and SVBRDF capture under global illumination effects is still difficult.

3. Method

Figure 1 shows our exploratory reconstruction pipeline based on differentiable path tracing. Starting with a set of images captured from known viewpoints and under known illumination, we propose a procedure which first constructs an initial coarse estimate of object geometry using existing methods, and then alternates between optimizing this geometry and a mesh colors [48] spatially-varying material using gradient descent with a differentiable path tracer. Our proposed procedure has multiple sub-components; the remainder of this section motivates and describes each in detail.

3.1. Input

The input to our pipeline is a set of images of the target object plus the scene lighting, represented either by an HDR environment map or a set of point lights. Images are captured from known poses and under a known lighting configuration as might be captured by a light stage [40] or box [13], from a multi-view stereo setup with known camera/light offset [29], or as frames from a low-dynamic-range video sequence from a hand-held cell phone with known environment lighting. Typically, the greater the number of views or frames, the higher the quality of reconstruction.

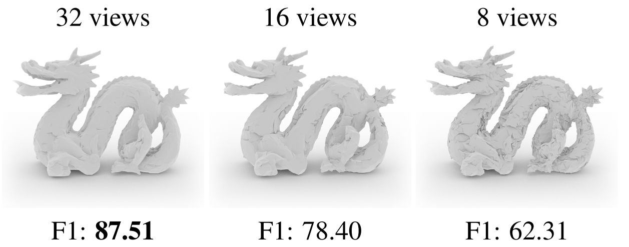

Figure 2. Degradation of reconstruction quality with decreasing number of input views. As the number of views to constrain the optimization decreases, more noise appears in the geometry reconstruction. F1 score is computed with a tolerance of 0.01.

Figure 2 shows the effects of number of input views on reconstruction quality.

For scene lighting input, we use point lights to reconstruct objects in simulation, and environment maps to reconstruct objects in real-world scenes. In the supplemental material, we investigate the effect of illumination model (environment vs. point lights) on reconstruction quality. We find that reconstruction is more accurate under point illumination than environment map illumination. However, environment maps better model real-world scenes.

We only consider input image pixels covering the object and mask away the image background. Masks can be created manually or via classic or machine-learned image segmentation. In simulation, we compute binary foreground masks via a white albedo render of the relevant scene geometry. For real-world imagery, we use the *X101-FPN* model from *detectron2*, pretrained on the COCO dataset, to perform instance segmentation [44].

Similarly, image-space masks for separate materials that appear in the same object (e.g., distinguishing a plastic bottle from a metal bottle cap) are helpful for controlling the material optimization landscape. These can be computed in a variety of ways: manually, for finest precision, or clustering image pixels by their RGB color or normalized intensity.

3.2. Geometry Optimization

Given the above inputs, our approach is to alternate between optimizing the geometry and material of the reconstruction. We detail the geometry phase of this alternating minimization scheme: the representation, initialization, and a multi-step approach to perform gradient-based refinement on geometry while controlling its resolution and quality.

Representation We use a triangle mesh to represent object geometry. First, it provides local control over geometry, allowing for optimization to locally capture fine detail. Second, it supports coarse-to-fine refinement, which our optimization schedule heavily exploits. Third, it naturally accommodates SVBRDF specification via a per-mesh-facet representation. Finally, it facilitates highly-optimized ray-surface intersection, which forms the bulk of path tracing's computational cost. The major limitation of a mesh, as opposed to an implicit representation, is that it is more difficult to change topology during optimization. As we will see later in this section, periodic remeshing during optimization can overcome this difficulty.

Initialization Since our optimization is based on gradient-based local optimization, it is important to start with an initial mesh that captures large-scale topological features to place the optimizer in the right basin of attraction. Possible initialization strategies are to start with a simple proxy geometry, e.g., spheres or boxes, which can be used in any setting but may require significant hand-tuning to work well; or to leveraging existing bottom-up reconstruction methods—these give more accurate initial results but may make assumptions about the underlying scene. We have experimented with the multi-view stereo pipeline COLMAP [36], voxel carving, and 'sphere clouds,' which we detail in the supplemental material.

Figure 3 illustrates the behavior of different initalization strategies when refining them with our suggested procedure. MVS produces the highest quality initializations and therefore the best reconstruction, but requires a large number of input views (>100). Voxel carving operates more reliably under a range of camera views (as few as six) at the cost of some geometric detail. The sphere cloud approach, while general-purpose, is least accurate. In the supplemental material, we also explore refining initial geometry produced by a deep-learning based reconstruction method.

Vertex optimization Given the initial mesh geometry, the first (and core) step of the geometry phase is to optimize mesh vertex positions $\mathbf{x}$ via gradient descent with respect to a mean-squared-error loss:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{M}_i^t \cdot \mathcal{T}_i - \mathcal{M}_i^r \cdot R(\mathbf{x}, c_i))^2$$

where $\mathcal{T}_i$ is the ith target image, c_i is the ith camera pose, $\mathcal{M}_i^t$ is the ith target mask, $\mathcal{M}_i^r$ is the ith mask of the current reconstructed object, and R is a differentiable physically-based rendering function. We use the differentiable path tracer of Li et al. [18], as it provides gradients of output pixels with respect to input geometry.

Subdivision To avoid poor local optima, we found that coarse-to-fine optimization works well. We begin with a low-resolution initial mesh, optimize its vertices, and then increase mesh resolution once this optimization converges. To increase resolution, we subdivide every triangular face into four by splitting each edge at the midpoint. This approach helps balance the granularity of geometric and mate-

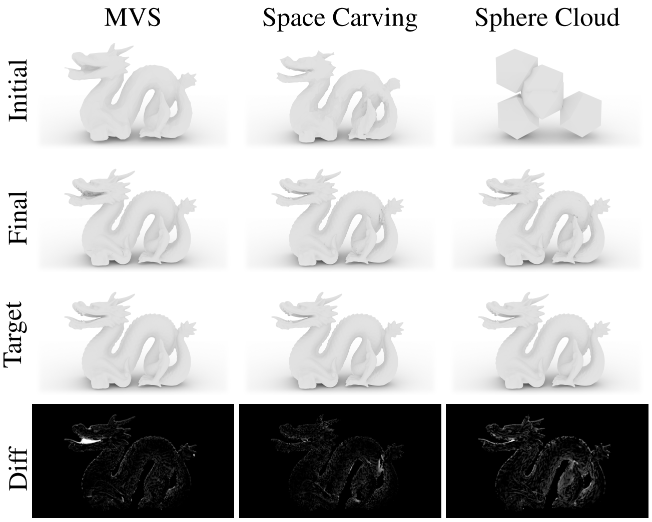

Figure 3. Reconstruction results on different initializations. Top row: initializations produced by multi-view stereo *(Top Left)*, voxel carving *(Top Middle)*, and 'sphere clouds' *(Top Right)*. Second row: results of geometry-only optimization starting from each initialization. Bottom row: difference (2× magnified) between the final reconstruction and the target input image.

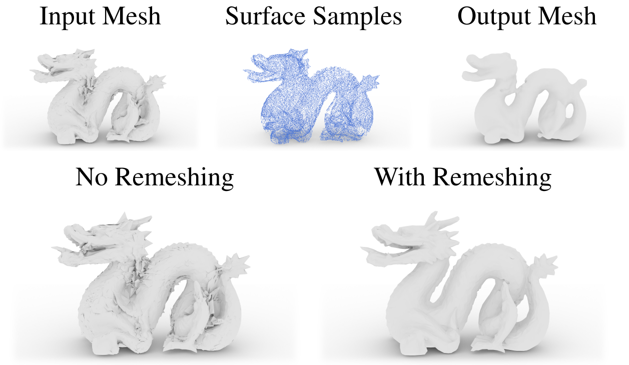

Figure 4. Effect of the remeshing step. *(Left)* A degraded mesh mid-optimization. *(Middle)* Point samples on the exterior surface. *(Right)* Poisson surface reconstruction on points. Artifacts disappear and fine detail can be recovered by subsequent optimization. The genus of the object changes from 0 to 1. *(Bottom)* Final outputs. Self-intersecting geometry degrades the reconstruction and hinders further optimization. Both optimizations are initialized as sphere clouds to emphasize behavior of this step.

rial detail at each iteration, preventing geometry from compensating for missing texture detail and preventing texture from 'baking in' geometric detail.

Simplification When increasing mesh resolution, we would ideally control the size and stability of the parameter space, as well as the efficiency of optimization, by only adding vertices to mesh regions that require finer detail. Thus, we follow subdivision with mesh simplification. If, after subdivision, the number of faces exceeds a threshold $\mathcal{D}$, we decimate the mesh using quadric error simplification [6] until it has $\mathcal{D}$ faces remaining. $\mathcal{D}$ is initially set to twice the number of faces, and increases by half the number of initial faces after every simplification step. While Xu et al. [46] seek to avoid vertices distributed non-uniformly across the surface of the mesh, we find that allowing this behavior facilitates estimation of more complex geometries.

Remeshing Due to Monte Carlo rendering noise and the limited number of target images, gradient descent on vertex positions can result in artifacts from which it cannot recover (e.g., excess triangles in the interior of the mesh). To rectify this problem, we re-generate the mesh by point-sampling its surface via raycasting from the input viewpoints, followed by a Poisson surface reconstruction [15] on the resulting point cloud (Figure 4, bottom). This strategy also allows our input mesh to be less sensitive to the target object's genus, i.e., the remeshing step can open holes in the optimized mesh to match the target (Figure 4, top).

Shape From Shading Since we use multiple views, it is possible that the quality of geometry reconstruction is mainly due to the large percentage of the target object which is seen in silhouette. We conduct an experiment to reconstruct geometry in the absence of silhouette edges and from shading only. The target object is a sphere with a divot that does not affect the shape's silhouette. Figure 5 shows reconstruction results for this case. Starting with an initial sphere mesh, our optimizer depresses the surface down to just short of the target depth, demonstrating that concave surface detail can be accurately captured. This behavior is critical to reconstruct surfaces whose concavities cannot be fully captured by silhouette-only initialization.

3.3. Material Optimization

In alternation with geometry optimization, we also optimize for the material of the object. The remainder of this section motivates and details each component of our material optimization stage.

Representation We represent material as a Torrance-Sparrow BRDF [39], a physically-based microfacet model commonly used in material acquisition research [9]. It is parameterized by a diffuse albedo and a specular roughness, traditionally represented by UV-mapped textures. However, UV maps are ill-suited for our setting as our mesh is constantly changing, and we would need to simultaneously optimize the UV surface parameterization with the contents of the texture image. Instead, we use mesh colors [48]: an adaptive-resolution extension of vertex colors. This is suitable for our optimization as it (a) provides an automatic coarse-to-fine level of detail which is tied to the underly-

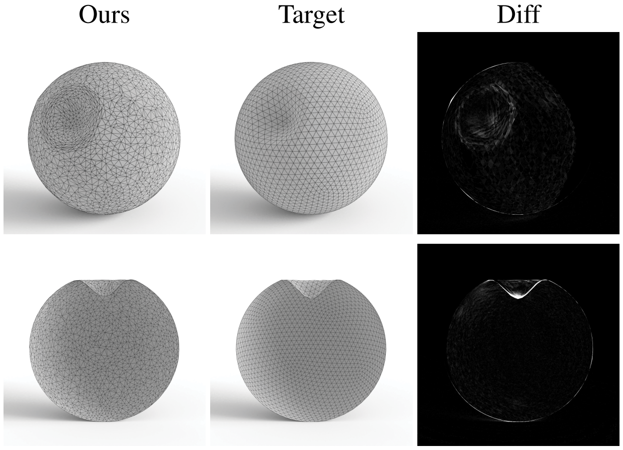

Figure 5. Reconstruction of a sphere-with-divot, starting from a sphere, using 5 cameras each with 4 light angles. With no silhouette cues, from only shading information, our optimization pushes down the divot to just short of the required depth, as demonstrated by the difference image (right, intensity magnified 10x).

ing triangle mesh geometry, (b) does not require a surface parameterization, and (c) is seamless by construction.

We modify the path tracing framework of Li et al. [18] to support mesh colors. A mesh colors texture is stored in a 1-D array $\mathbf{a}$, the size of which is determined by the number of triangles in the mesh and an integer resolution level r. Given the barycentric coordinates (α, β) of a point in triangle number t, the texel for that point is

$$\begin{aligned} \mathbf{m}(r, t, \alpha, \beta) &= \mathbf{a}[k] \\ k &= \frac{t \cdot (r+1) \cdot (r+2) + i \cdot (2r - i + 3)}{2} + j \\ (i, j) &= \lfloor r \cdot (\alpha, \beta) \rfloor \end{aligned}$$

This storage scheme duplicates edge and vertex detail for parallelism at the slight cost of additional memory. We use finite differences to compute derivatives.

Initialization We optimize texture maps in a coarse-to-fine manner. For the first five optimization cycles, we optimize for a single spatially invariant diffuse color and specular color per material. Using a low-degree-of-freedom material representation while the geometry is initially refined prevents the material from 'baking in' appearance effects from small-scale geometry. To distinguish between materials, we use per-material image-space masks to segment distinct materials. To correct for any inconsistencies in clustering from view to view, we determine which vertices belong to each material cluster by counting which material they are assigned to most often in image-space.

Every time the mesh changes topology (i.e., after a remeshing step), we re-optimize from a neutral gray color to avoid bias towards previous errors in geometry or texture.

The texture loss is:

$$\mathcal{L}(\mathbf{x}, \mathbf{m}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{s} (\mathcal{M}_{i,j} \cdot \mathcal{T}_i - \mathcal{M}_{i,j} \cdot R(\mathbf{x}, \mathbf{m}, c_i))^2$$

where colors or per-triangle texels $\mathbf{m}$ are optimizable, and $\mathcal{M}_{i,j}$ is the jth material mask for the ith target image.

Spatially-varying Diffuse After the first five optimization cycles, we expand the parameter space to allow a spatially-varying diffuse material, initialized from the last spatially invariant diffuse color, while keeping the single specular value. We estimate areas where we expect specular highlights to occur by rendering the geometry as a perfect mirror and thresholding bright areas. Then we mask out these bright regions from renders. This removes gradients where specular highlights occur. As highlights vary from target frame to frame and have high radiance, they otherwise tend to 'bake' into spatially-varying diffuse texture.

Spatially-varying Specular For the last optimization cycles, we optimize for a spatially-varying specular material. As low-variance (and often constant) specular maps are a common artistic choice, we add a variance penalty on the specular mesh colors $\mathbf{m}_s$: $\mathcal{L}(\mathbf{x}, \mathbf{m}) = \frac{1}{n} \sum_{i=1}^{n} (\mathcal{M}_i \cdot \mathcal{T}_i - \mathcal{M}_i \cdot R(\mathbf{x}, \mathbf{m}, c_i))^2 + \lambda \text{Var}(\mathbf{m}_s)$.

Effects of Global Illumination Previous differentiable renderering-based reconstruction methods do not use global illumination. We justify the use of global illumination by investigating the effect of diffuse interreflection on material estimation. We set up a virtual scene consisting of three intersecting planes, each with a constant diffuse albedo: red, blue, and white. The geometry is known, and we optimize a SVBRDF material for this geometry given a rendered target. We compare results to a version in which the renderer uses only one-bounce illumination. Figure 6 shows the results. Optimization with global illumination correctly reconstructs the ground-truth albedos, while the one-bounce version explains the purplish floor (caused by color bleeding) by baking this color into the ground plane's albedo.

3.4. Optimization Details

Our optimization procedure alternates between solving for geometry and material, switching once the loss has converged. Optimization proceeds until the loss stops improving between successive cycles.

4. Results

In this section, we explore reconstructing complex object geometry and material in simulation and from real-world input. All results were produced on desktops with an AMD Ryzen 2700X and an NVIDIA GTX 1080Ti.

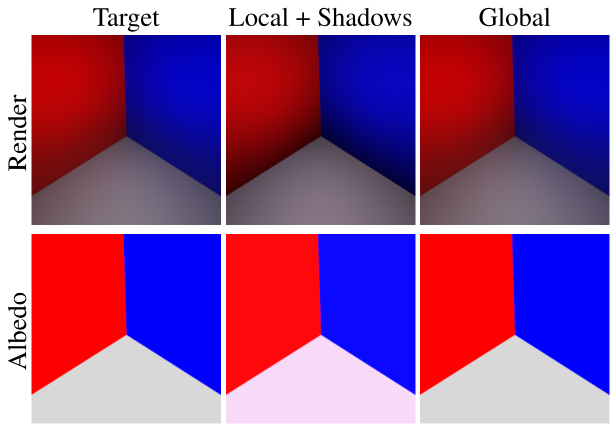

Figure 6. Material optimization using local vs. global illumination. Top: each strategy's output—all quite similar. Bottom: unlit albedo of each strategy. The local case *(Bottom Middle)* 'bakes' purple diffuse interreflection into the floor's albedo, while a differentiable path tracer disambiguates this effect *(Bottom Right)*.

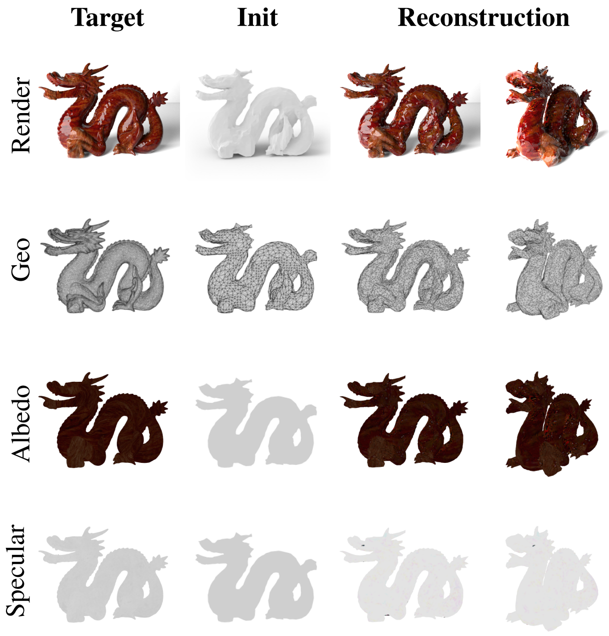

Figure 7. Reconstruction of the Stanford dragon inside a Cornell Box, with a glossy wood material. The image resolution was 128 × 128. Col. 2 shows initializations; Col. 4 shows a novel view. Although the geometry reconstruction exhibits dense triangle clusters, it is intersection-free: clusters are caused by our adaptive remeshing, which concentrates vertices in high detail areas. Re-renderings were produced using the Blender Cycles renderer..

4.1. Reconstructing Simulated Objects

We test our findings on the task of reconstructing 3D objects with known geometry and material in simulation, using meshes varying in genus and texture patterns.

Table 1. Quantitative results for Figure 7. We report Chamfer distance and F1 scores between initialization/target (condition 1) and reconstruction/target geometries (condition 2). We report PSNR *(Left)* and SSIM *(Right)* between initialization/target and reconstruction/target renders.

Object	**Cond.**	**Chamfer**	**F1**	**Full**	**Diffuse**	**Specular**
Dragon	Init.	0.191	44.61	9.82, 0.69	10.44, 0.68	27.37, 0.70
	Recon.	**0.149**	**87.51**	**32.19**, **0.91**	**33.89**, **0.87**	**43.78**, **0.98**
Armadillo	Init.	0.158	47.97	12.16, 0.73	12.34, 0.72	28.94,0.69
	Recon.	**0.147**	**98.72**	**29.25**, **0.89**	**26.58**, **0.84**	**36.83**,**0.94**
Buddha	Init.	0.172	47.97	15.20, 0.85	15.98, 0.84	22.47, 0.72
	Recon.	**0.155**	**87.68**	**31.96**, **0.96**	**33.80**, **0.94**	**32.24**, **0.88**

We use 32 cameras distributed on a Fibonacci sphere surrounding the object within a Cornell box, with two light position variations per view; one light is aligned with the camera and another is placed at a constant offset of 1 unit in the camera's tangent direction. Using multiple light positions is critical to distinguish between geometric surface details, diffuse albedo, and specular highlights. Figure 7 shows qualitative results (with additional results in supplemental material). Table 4.1 shows numerical results.

4.2. Reconstructing Real-World Objects

Most existing real-world methods requires objects be photographed in HDR in a controlled environment, e.g., in a dark room with a few fixed lights. Here, we use differentiable path tracing to address a more challenging ill-posed scenario: an object casually-captured outdoors with a smartphone video and a 360 camera HDR. This is a difficult setting for reconstruction as we assume no constraints on illumination and only require a coarse estimate of scene lighting and a sparse number of views with estimated poses.

Data capture We capture input views and lighting using consumer hardware. For input views, we use a Pixel 3A phone to record video of an object while walking around it, using the built-in camera app and the H.264/AVC video format. For lighting capture, we place a Insta360 ONE camera in the same position as the object, take an exposure bracket of the environment, and fuse these into a HDR environment map. The supplemental material shows our setup.

Reconstruction methodology We sample every 60th frame of the video, use COLMAP to create the geometry initialization and estimate each frame's associated camera pose, and align the environment map manually. To account for different camera responses between real and simulated cameras, we optimize for an environment map brightening factor during the first round of texture estimation.

Results Fig. 8 shows results from our real-world reconstruction experiment. Despite the relatively unconstrained

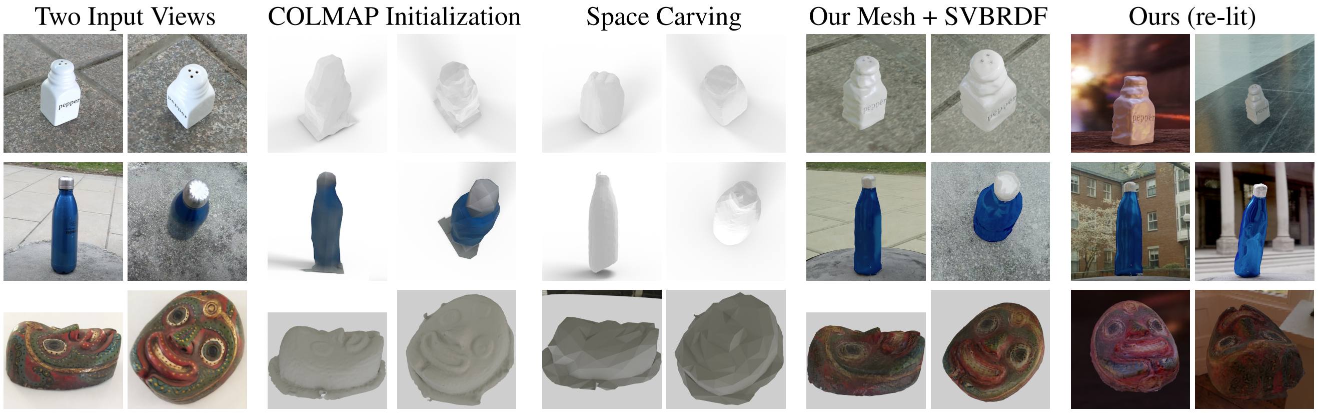

Figure 8. Results from real world capture. We show two video frames from the smartphone camera *(Col. 1)*, and the geometry initialization *(Col. 2)*. Our geometry estimations compare favorably against their initializations and space carving reconstruction *(Col. 3)*: note the entire upper half of the pepper shaker. We show our final reconstruction rendered under captured *(Col. 4)* and novel illumination *(Col. 5)*.

capture set up, our prototype recovers spatially varying texture detail, and a more accurate geometry prediction as a refinement to the COLMAP reconstructions. These reconstructions do not recover the same level of detail that they do in simulated environments; we expect some inaccuracy in both geometry and SVBRDF predictions due to compounding error in COLMAP's camera calibration and our environment map alignment. Improving these parts of scene set-up is required to recover finer-scale detail.

5. Discussion

Our investigations assume known camera positions and some estimate of lighting conditions. This constraint might be relaxed in the future with simultaneous optimization of camera position, texture, geometry, and lighting. In addition, our optimizations require a reasonable initialization so that differentiable path tracing can produce meaningful gradients. As such, we see it as a refinement technique. Our method is comparable in runtime to a dense COLMAP initialization, taking 3–5 hours depending on the resolution and number of views. This is expected due to the natural expense of path tracing and the added expense of backpropagating gradients. Recent ray tracing hardware may offer improvements to differentiable path tracing speed. Finally, our current mesh colors representation is not mipmapped or anisotropically filtered, and these additions would improve texture estimation across pixels of varied depth.

One avenue of future work is more complex materials, especially those exhibiting ideal reflection/refraction, subsurface scattering, or volumetric effects. Another possible area of investigation is to raise the ceiling on object reconstruction size, broadening the scope of possible targets from single small objects to larger scenes like rooms and city landscapes. The sparsity of the mesh representation we use lends itself to such reconstruction tasks. This would require a reevaluation of lighting environment and viewing angle assumptions. Another possible direction is how best to combine learned neural network priors with our methodology to leverage the advantages of both. For instance, learned priors can encode artistic intent and class-specific patterns, while our approach recovers fine-scale geometry details. A hybrid approach could lead to flexible physically-accurate reconstruction systems.

6. Conclusion

We have investigated how to use differentiable path tracing to jointly estimate the shape and material of a 3D object under known lighting conditions from a series of target images. Starting from a coarse geometry initialization, we alternate between texture and geometry steps and gradually increase the parameter space for optimization. We motivate pipeline stages with several experiments, and show that optimizing over global illumination effects can help handle interreflections and self shadows in reconstruction. We find that optimization via a differentiable path tracer is a promising avenue of research for shape and material reconstruction in unconstrained settings. Finally, we show that our method can refine results on real-world data from largely unconstrained capture setups using smartphone videos.

Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful and helpful suggestions. We also thank Prof. Min H. Kim for discussions about object capture and reconstruction, and Prof. Tzu-Mao Li for discussions about differentiable path tracing. This research was supported by GPU donations from NVIDIA and computer donations from Valve Corporation.

References

[1] D. Azinović, T.-M. Li, A. Kaplanyan, and M. Nießner. Inverse path tracing for joint material and lighting estimation. In *CVPR*, 2019. 2

[2] C. Che, F. Luan, S. Zhao, K. Bala, and I. Gkioulekas. Towards learning-based inverse subsurface scattering. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2020. 2

[3] W. Chen, J. Gao, H. Ling, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances In Neural Information Processing Systems*, 2019. 2

[4] A. Dib, G. Bharaj, J. Ahn, C. Thebault, P.-H. Gosselin, and L. Chevallier. Face reflectance and geometry modeling via differentiable ray tracing, 2019. 2

[5] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Transactions on Graphics (TOG)*, 38(4):134, 2019. 2

[6] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 209–216, 1997. 5

[7] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman. Unsupervised training for 3d morphable model regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[8] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, June 2010. 2

[9] D. Guarnera, G. C. Guarnera, A. Ghosh, C. Denk, and M. Glencross. BRDF representation and acquisition. In *Computer Graphics Forum*, volume 35, pages 625–650. Wiley Online Library, 2016. 5

[10] P. Henderson and V. Ferrari. Learning single-image 3D reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, 2019. 2

[11] P. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. Huang. Deepmvs: Learning multi-view stereopsis. *CoRR*, abs/1804.00650, 2018. 2

[12] Y. Jiang, D. Ji, Z. Han, and M. Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization, 2019. 2

[13] K. Kang, C. Xie, C. He, M. Yi, M. Gu, Z. Chen, K. Zhou, and H. Wu. Learning efficient illumination multiplexing for joint capture of reflectance and shape. *ACM Trans. Graph.*, 38(6), Nov. 2019. 3

[14] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[15] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70. Eurographics Association, 2006. 5

[16] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, July 2000. 2

[17] F. Langguth, K. Sunkavalli, S. Hadap, and M. Goesele. Shading-aware multi-view stereo. In *European Conference on Computer Vision*, pages 469–485. Springer, 2016. 2

[18] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 1, 2, 4, 6

[19] Z. Li, K. Sunkavalli, and M. Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. *CoRR*, abs/1804.05790, 2018. 2

[20] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia 2018 Technical Papers*, page 269. ACM, 2018. 2

[21] Y. Lin, P. Peers, and A. Ghosh. On-site example-based material appearance acquisition. *Computer Graphics Forum*, 38(4):15–25, 2019. 2

[22] S. Liu, T. Li, W. Chen, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 2

[23] S. Liu, S. Saito, W. Chen, and H. Li. Learning to infer implicit surfaces without 3d supervision. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8293–8304. Curran Associates, Inc., 2019. 2

[24] S. Lombardi and K. Nishino. Radiometric scene decomposition: Scene reflectance, illumination, and geometry from rgb-d images. *2016 Fourth International Conference on 3D Vision (3DV)*, Oct 2016. 3

[25] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4):65:1–65:14, July 2019. 2

[26] M. M. Loper and M. J. Black. Opendr: An approximate differentiable renderer. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *ECCV*, 2014. 1, 2

[27] I. Mallett, L. Seiler, and C. Yuksel. Patch textures: Hardware implementation of mesh colors. In *High-Performance Graphics (HPG 2019)*. The Eurographics Association, 2019. 2

[28] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 2

[29] G. Nam, J. H. Lee, D. Gutierrez, and M. H. Kim. Practical SVBRDF acquisition of 3d objects with unstructured flash photography. *ACM Trans. Graph.*, 37(6), Dec. 2018. 3

[30] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), Nov. 2019. 1, 2

[31] J. J. Park, A. Holynski, and S. Seitz. Seeing the world in a bag of chips, 2020. 3

[32] F. Petersen, A. H. Bermano, O. Deussen, and D. Cohen-Or. Pix2vex: Image-to-geometry reconstruction using a smooth differentiable renderer. *CoRR*, abs/1903.11149, 2019. 2

[33] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Pytorch3d. https://github.com/facebookresearch/pytorch3d, 2020. 2

[34] H. Rhodin, N. Robertini, C. Richardt, H.-P. Seidel, and C. Theobalt. A versatile scene model with differentiable visibility applied to generative pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 1, 2

[35] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[36] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 4

[37] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 519–528, Washington, DC, USA, 2006. IEEE Computer Society. 2

[38] V. Sitzmann, M. Zollhöfer, and G. Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations, 2019. 2

[39] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of The Optical Society of America*, 57, 09 1967. 5

[40] B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, and P. Debevec. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Trans. Graph.*, 32(4), July 2013. 3

[41] J. Valentin, C. Keskin, P. Pidlypenskyi, A. Makadia, A. Sud, and S. Bouaziz. Tensorflow graphics: Computer graphics meets deep learning. 2019. 1, 2

[42] M. Weinmann, F. Langguth, M. Goesele, and R. Klein. Advances in Geometry and Reflectance Acquisition. In A. Sousa and K. Bouatouch, editors, *EG 2016 - Tutorials*. The Eurographics Association, 2016. 2

[43] C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *CVPR 2011*, pages 969–976. IEEE, 2011. 2

[44] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4

[45] R. Xia, Y. Dong, P. Peers, and X. Tong. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Transactions on Graphics*, 35(6), December 2016. 3

[46] Z. Xu, S. Bi, K. Sunkavalli, S. Hadap, H. Su, and R. Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019. 2, 5

[47] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 2

[48] C. Yuksel, J. Keyser, and D. H. House. Mesh colors. *ACM Trans. Graph.*, 29(2):15:1–15:11, Apr. 2010. 2, 3, 5

[49] C. Zhang, L. Wu, C. Zheng, I. Gkioulekas, R. Ramamoorthi, and S. Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38(6), 2019. 2