Reconstructing Translucent Objects using Differentiable Rendering

Xi Deng
Cornell University
Ithaca, NY, USA
xd93@cornell.edu

Fujun Luan
Adobe Research
San Jose, CA, USA
fluan@adobe.com

Bruce Walter
Cornell University
Ithaca, NY, USA
bruce.walter@cornell.edu

Kavita Bala
Cornell University
Ithaca, NY, USA
kb@cs.cornell.edu

Steve Marschner
Cornell University
Ithaca, NY, USA
srm@cs.cornell.edu

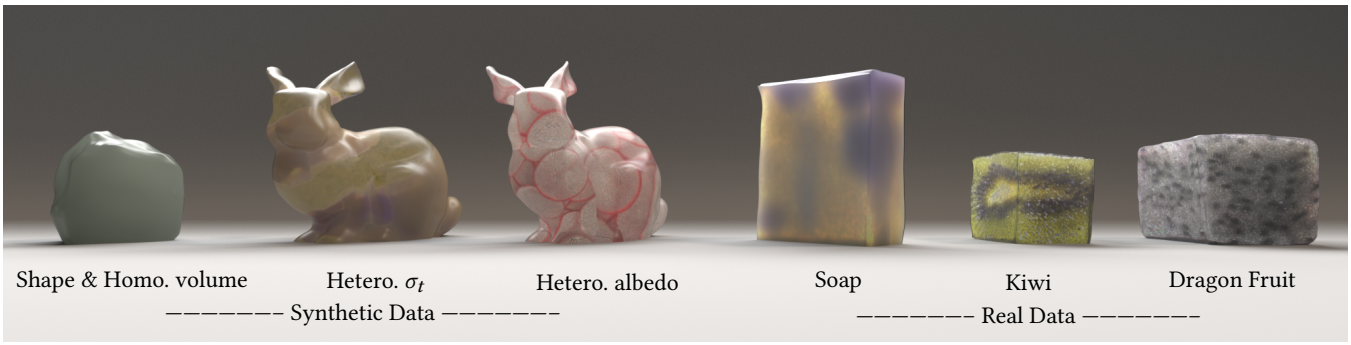


Figure 1: Reconstruction results rendered in global illumination for both synthetic (left three objects) and real data (right three objects). For the synthetic data, we show jointly reconstructing the shape and subsurface scattering material of a bumpy object (first from left); a spatially varying extinction coefficient texture (left bunny) and a spatially varying single scattering reflectance texture (right bunny). On real data, we show reconstruction of a slice of soap and cut cubes of kiwi and dragonfruit.

ABSTRACT

Inverse rendering is a powerful approach to modeling objects from photographs, and we extend previous techniques to handle translucent materials that exhibit subsurface scattering. Representing translucency using a heterogeneous bidirectional scattering-surface reflectance distribution function (BSSRDF), we extend the framework of path-space differentiable rendering to accommodate both surface and subsurface reflection. This introduces new types of paths requiring new methods for sampling moving discontinuities in material space that arise from visibility and moving geometry. We use this differentiable rendering method in an end-to-end approach that jointly recovers heterogeneous translucent materials (represented by a BSSRDF) and detailed geometry of an object (represented by a mesh) from a sparse set of measured 2D images in a coarse-to-fine framework incorporating Laplacian preconditioning for the geometry. To efficiently optimize our models in the presence of the Monte Carlo noise introduced by the BSSRDF integral, we introduce a dual-buffer method for evaluating the L2 image loss. This efficiently avoids potential bias in gradient estimation due to the correlation of estimates for image pixels and their derivatives and enables correct convergence of the optimizer even when using low sample counts in the renderer. We validate our derivatives by comparing against finite differences and demonstrate the effectiveness of our technique by comparing inverse-rendering performance with previous methods. We show superior reconstruction quality on a set of synthetic and real-world translucent objects as compared to previous methods that model only surface reflection.

CCS CONCEPTS

• **Computing methodologies** → **Rendering**; **Ray tracing**.

KEYWORDS

ray tracing, subsurface scattering, appearance acquisition, differentiable rendering

ACM Reference Format:
Xi Deng, Fujun Luan, Bruce Walter, Kavita Bala, and Steve Marschner. 2022. Reconstructing Translucent Objects using Differentiable Rendering. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings), August 7–11, 2022, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3528233.3530714

1 INTRODUCTION

Accurate reconstruction and efficient rendering of objects with highly scattering translucent materials is challenging and important in the fields of digital prototyping, architecture, entertainment, and even the biological sciences. This kind of material, with subsurface scattering, is ubiquitous in real life, ranging from organic tissues like fruits, vegetables, meat, juice and milk, to vast categories of

SIGGRAPH '22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada

ACM ISBN 978-1-4503-9337-9/22/08…$15.00
https://doi.org/10.1145/3528233.3530714

inorganic materials such as gems, soap, ceramics and plastics. They share common features like heterogeneity, surface microstructure, and short scattering mean free paths.

Subsurface scattering effects can be described by the *Bidirectional Scattering-Surface Reflectance Distribution Function* (BSSRDF), which directly models the light transport between pairs of surface points [Nicodemus et al. 1977]. The BSSRDF encodes non-local illumination due to the underlying volumetric light transport, and it is the basis for various methods to compactly represent and efficiently render translucent objects [Chen et al. 2004; d'Eon and Irving 2011; Frisvad et al. 2014; Goesele et al. 2004; Habel et al. 2013; Peers et al. 2006; Song et al. 2009; Tong et al. 2005; Vicini et al. 2019]. These works have mainly assumed known geometry, and many rely upon special lighting setups. However, the geometry of translucent objects can be difficult to measure by standard computer vision methods, and the need for special setups limits the applicability of translucent object capture. Jointly recovering the geometry and appearance of translucent objects under ordinary lighting in a robust way remains an unsolved problem, especially from sparse views.

In this paper we propose a way forward by applying an inverse rendering approach using a differentiable subsurface-scattering renderer. Building on the considerable recent progress in differentiable rendering theory, algorithms and systems [Bangaru et al. 2020; Li et al. 2018; Loubet et al. 2019; Nicolet et al. 2021; Nimier-David et al. 2020, 2019; Vicini et al. 2021; Zhang et al. 2021a, 2020, 2021b], we extend the state of the art to support BSSRDF materials. Previous differentiable renderers support translucency only by representing the material as a participating medium and using costly volumetric path tracing, typically requiring thousands of samples per pixel, per optimization iteration, for an acceptable amount of noise in the estimated gradients. Ours, in contrast, uses the BSSRDF to model translucency for object capture. This empirical model approximates large-scale simulations of volume scattering by a physically-based diffusion model with closed-form solutions, which allows us to render and optimize using significantly fewer samples per iteration.

By fitting renderings of mesh geometry with a spatially varying BSSRDF material to photographs in an end-to-end optimization that finds both the surface geometry and the material properties, we recover models that match real objects under lighting conditions similar to the capture. Using the relevant goal—similarity of final renderings—to drive the fitting process, and recovering geometry and material jointly, means that our approach gets the best results out of the representation without needing the data to perfectly reveal either the geometry or the material in isolation. Including translucency in inverse rendering is also important because attempting to recover translucent geometry using an opaque shading model is problematic, as shown by our comparisons.

Concretely, our contributions include:

- A new formulation of Monte Carlo differentiable rendering for BSSRDF models that handles geometric discontinuities in an unbiased way.
- A new analysis-by-synthesis optimization pipeline that enables high-quality joint reconstruction of heterogeneous translucent materials and mesh-based object geometry.
- A dual-buffer solution for estimating gradients of an L2 loss that ensure better convergence even at low sample counts when the path distribution used by the differentiable renderer depends on the scene parameters.

Through validations and evaluations of individual steps, we demonstrate that our method significantly advances 3D reconstruction of translucent objects using low-cost handheld acquisition setups, which can be beneficial for many applications. Our results show that it is important to include translucency in the model when using inverse rendering on surfaces that exhibit translucency, even if only the shape is desired, since methods assuming an opaque surface will converge to less accurate geometry. Code and data for this paper are at https://github.com/joyDeng/InverseTranslucent.

2 RELATED WORK

Subsurface Scattering. Subsurface scattering is a mechanism of volumetric light transport in which light that penetrates the surface of a translucent object is scattered inside the material before either being absorbed or leaving the material at a different location. This process is generally modeled using the radiative transfer equation (RTE) [Chandrasekhar 1960] and involves long sequences of scattering interactions within a medium that fills the object's interior. In rendering, this scattering can be simulated directly; existing techniques include volumetric path tracing and photon mapping algorithms [Novák et al. 2018]. Although it can produce accurate results, the volumetric approach is computationally expensive due to the large number of subsurface scattering events, especially for highly scattering materials, and the BSSRDF is an alternative approach that is often much faster.

For highly scattering materials, the RTE can be usefully approximated by a diffusion equation, and many BSSRDF models have been proposed based on approximate analytical diffusion solutions. Jensen et al. [2001] presented a practical dipole model to compactly represent homogeneous subsurface scattering materials, and later work has proposed more accurate though generally more complex models [d'Eon and Irving 2011; Frisvad et al. 2014; Vicini et al. 2019].

To add heterogeneity, early research extended the Jensen et al. dipole model to heterogeneous skin BSSRDFs by fitting dipoles for each surface point [Donner et al. 2008; Tariq et al. 2006] or per region [Ghosh et al. 2008; Weyrich et al. 2006]. Tong et al. [2005] improved the representation for a large class of *quasi-homogeneous* materials. Goesele et al. [2004] decomposed general BSSRDFs into a local filtering kernel and a low-res global term. Fuchs et al. [2005] fitted a sum of radial exponential kernels and represented spatially-varying parameters in textures. Peers et al. [2006] used a data-driven representation and factorized spatial variations via a modified matrix factorization algorithm. Donner et al. [2009] presented an empirical diffusion model derived from a large-scale Monte Carlo simulation. Song et al. [2009] decomposed the BSSRDF into a two-dimensional collection of scattering profiles to enable intuitive editing while maintaining consistency, and we follow their general approach in this paper.

Inverse scattering and Differentiable Rendering. Many projects have recovered parameters of media or BSSRDF models by solving inverse problems. Some techniques rely on single-scattering approximations [Fuchs et al. 2007; Hawkins et al. 2005; Narasimhan et al. 2006], which are appropriate for optically thin media; others on diffusion-based models [Dong et al. 2014; Jensen et al. 2001;

Mukaigawa et al. 2009; Munoz et al. 2011; Papas et al. 2013; Song et al. 2009; Wang et al. 2008], suitable for optically thick media.

In more general settings where these approximations do not apply, one can resort to numerical optimization guided by differentiable models of light transport. This approach has been used to solve for specific transport effects in contexts such as volumetric scattering [Che et al. 2020; Gkioulekas et al. 2016, 2013], cloth rendering [Khungurn et al. 2015], volume prefiltering [Zhao et al. 2016], translucent fabrication [Elek et al. 2017; Nindel et al. 2021; Rittig et al. 2021; Sumin et al. 2019], human teeth modeling [Velinov et al. 2018], material and lighting estimation [Azinovic et al. 2019], cloud modeling [Levis et al. 2015], and time-gated rendering [Wu et al. 2021; Yi et al. 2021]. Many of these methods are specialized to specific types of scene parameters and most of them avoid discontinuities.

Accommodating discontinuities in general-purpose differentiable rendering is challenging and generally requires additional boundary integrals to account for moving discontinuities. Li et al. [2018] proposed to solve this problem by explicitly sampling the discontinuities via a Monte Carlo edge-sampling method, yielding unbiased estimates of the boundary integrals. Zhang et al. [2020, 2019, 2021b] further generalized the idea to handle participating media and path-space rendering. Loubet et al. [2019] proposed an approximate algorithm that avoids edge sampling through reparameterization, achieving lower variance at the cost of increased bias. Bangaru et al. [2020] then extended this technique using the divergence theorem to convert the boundary integral into an area integral for unbiased estimation. Nimier-David et al. [2019] introduced the Mitsuba 2 system that enables general-purpose differentiation with millions of parameters on GPU, with follow-up improvements on efficient reverse-mode differentiation schemes [Nimier-David et al. 2020; Vicini et al. 2021]. Finally, recent work [Zeltner et al. 2021; Zhang et al. 2021a] has investigated specialized Monte Carlo sampling strategies to reduce the variance of the derivative estimate.

These methods can handle translucent materials in principle, by representing them as scattering volumes, but in practice they cannot efficiently handle joint optimization of geometry and heterogeneous translucent materials, due to high variance in the estimated derivatives of volumetric path tracing integrals. In this work, we focus on heterogeneous translucent materials that can be represented by diffusion-based BSSRDF models, and compute end-to-end derivatives while allowing for geometric discontinuities caused by shadow edges and silhouettes. Unlike volumetric path tracing that relies on Monte Carlo sampling of path integrals, these analytical models are computationally efficient, and we show that our derivative estimates work in practice to solve inverse rendering problems with heterogeneous translucent materials.

3 OVERVIEW

Our main goal is to accurately reconstruct shape and heterogeneous translucent materials from photos, producing a physically-based model that can be directly used in a conventional renderer. In this section we explain the problem setup and the basic solution method; in Sections 4 and 5 we will provide further details about how we compute derivatives and use them in optimization.

Camera. We assume the input images are captured by known cameras and illumination comes from point sources at known locations. The camera and light position can be reconstructed using standard camera calibration or multi-view stereo tools (e.g., COLMAP [Schönberger and Frahm 2016]).

Lighting. Previous research on BSSRDF acquisition usually used numerous spatially varying lighting patterns together with camera grids, which is so time consuming that target objects like fruits and vegetables may dry out during the capture. While projecting special patterns on the surface is very revealing of the subtleties of BSSRDFs, we want to be able to render objects under relatively normal lighting to make capture easier for an end user, so in an inverse rendering setting we should be able to recover all that is needed without elaborate lighting conditions. Therefore we use point lighting and direct illumination for the capture process. Previous research [Luan et al. 2021; Nam et al. 2018] on reconstructing opaque materials used light sources co-located with the camera, which is easily accomplished when using a hand-held device and avoids casting shadows. However, co-located lights provide little information about translucency, which is better revealed by shadow and terminator edges. Therefore, we allow the point lights to move independently of the camera.

Appearance Model. We focus on reconstructing highly scattering translucent objects, and rendering subsurface scattering by volumetric path tracing is expensive in a participating medium under a rough dielectric surface due to long paths. Therefore, instead of differentiable volumetric path tracing [Zhang et al. 2021b], we use the BSSRDF and formulate the subsurface light transport problem as a surface integral:

$$L_o(\mathbf{x}_o, \omega_o) = \iint \rho_\mathrm{s}(\mathbf{x}_o, \mathbf{x}_i, \omega_o, \omega_i) L_i(\mathbf{x}_i, \omega_i) |\cos\theta_i| \,\mathrm{d}A \,\mathrm{d}\omega_i, \quad (1)$$

$$+ \int \rho_\mathrm{r}(\mathbf{x}_o, \omega_o, \omega_i) L_i(\mathbf{x}_o, \omega_i) |\cos\theta_i| \,\mathrm{d}\omega_i \quad (2)$$

where the outgoing radiance L_o is computed as a sum of contributions from subsurface light transport and surface reflection at $\mathbf{x}_o$; ρ_s is the BSSRDF model and ρ_r is the BRDF model. The contribution from subsurface light transport at $\mathbf{x}_o$ is a convolution of the incident illumination L_i and the BSSRDF ρ_s over incident position $\mathbf{x}_i$ and direction ω_i. Most models factor ρ_s into three terms:

$$\rho_\mathrm{s}(\mathbf{x}_o, \mathbf{x}_i, \omega_o, \omega_i) = S_w(\omega_o) R_d(\mathbf{x}_o, \mathbf{x}_i) S_w(\omega_i), \quad (3)$$

where $S_w(\omega_i)$ and $S_w(\omega_o)$ are transmission factors across the material boundary at $\mathbf{x}_i$ and $\mathbf{x}_o$, and $R_d(\mathbf{x}_o, \mathbf{x}_i)$ describes how much of the subsurface light that enters at $\mathbf{x}_i$ arrives at $\mathbf{x}_o$.

As discussed in Section 2, diffusion-based models are often used for R_d, and we use the model of Jensen et al. [2001], which models scattering at an infinite flat surface over a subsurface medium with extinction coefficient σ_t and scattering albedo α. To allow for heterogeneity in σ_t and α, we borrow an idea from Song et al. [2009] and further factorize the transmittance term as follows to allow for local parameters at $\mathbf{x}_i$ and $\mathbf{x}_o$:

$$R_d(\mathbf{x}_o, \mathbf{x}_i) = \sqrt{R_{\mathbf{x}_o}(\|\mathbf{x}_i - \mathbf{x}_o\|)} \sqrt{R_{\mathbf{x}_i}(\|\mathbf{x}_i - \mathbf{x}_o\|)}. \quad (4)$$

For the surface reflectance, we use a rough dielectric model with GGX distribution [Walter et al. 2007], which is parameterized by a roughness β and the index of refraction η of the surface. The parameters of BSSRDF and BRDF can be represented using texture maps on the surface or a uniform value.

Loss Function. We formulate our reconstruction problem as finding a parameter vector π, which includes the vertex positions of the mesh, the surface roughness, and the subsurface albedo α and extinction coefficient σ_t, by minimizing a loss function g that describes the difference between renderings $\mathbf{I}(\pi)$ and reference images $\mathbf{I}_{\text{ref}}$:

$$\arg\min_{\pi} g(\mathbf{I}(\pi)). \tag{5}$$

We measure the difference as a sum of squares, but instead of using the conventional L_2 loss we define the loss function as

$$g(\mathbf{I}(\pi)) = (\mathbf{I}_1(\pi) - \mathbf{I}_{\text{ref}})(\mathbf{I}_2(\pi) - \mathbf{I}_{\text{ref}}), \tag{6}$$

where $\mathbf{I}_1(\pi)$ and $\mathbf{I}_2(\pi)$ are two statistically independent Monte Carlo estimates of the noise-free image $\mathbf{I}(\pi)$. We show in Section 5 that our loss function efficiently provides the required unbiased estimate.

Differentiable Renderer. Efficiently solving (5) requires the gradient of g with respect to π, computed via the derivatives of pixel values with respect to each of the parameters. Previous work [Zhang et al. 2020] shows how to differentiate path tracing under the assumption of area lights and surface reflection modeled by the *Bidirectional Scattering Distribution Function* (BSDF) without ideal specular materials using a path integral framework.

4 DIFFERENTIABLE TRANSLUCENT RENDERING

In this section we extend the path integral framework of Zhang et al. [2020] and generalize the image derivative estimates to the setting of subsurface scattering.

4.1 Path Integral Framework with BSSRDF

In the path integral formulation [Veach 1998] a pixel value is:

$$I = \int_{\Omega(\pi)} f(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}}). \tag{7}$$

where $\bar{\mathbf{z}} = \mathbf{y}_0 \dots \mathbf{y}_n$ is the complete light path with its ends $\mathbf{y}_0$ and $\mathbf{y}_n$ at the camera and the light source respectively. We define two categories of path segment $\mathbf{y}_i\mathbf{y}_{i+1}$: subsurface segments going through translucent material and vacuum segments that pass through empty space. We refer to the two vertices on a subsurface segment as subsurface transmission vertices and other vertices as surface reflection vertices.

A measure on a path with length n is usually written as a product of area measures $\mu(\bar{\mathbf{z}}) = \prod_{i=1}^{n} \mathrm{d}A(\mathbf{y}_i)$, and all the paths with length n form an integration domain Ω_n; this domain depends on π so we write it as $\Omega_n(\pi)$. The space of paths with all possible lengths forms the path space $\Omega(\pi) = \cup_{n=1}^{\infty} \Omega_n(\pi)$.

The integrand of (7) is the contribution of a single light path,

$$f(\bar{\mathbf{z}}) = W_{\text{e}}(\mathbf{y}_0, \mathbf{y}_1) C(\bar{\mathbf{z}}) G(\bar{\mathbf{z}}) L_{\text{e}}(\mathbf{y}_n, \mathbf{y}_{n-1}), \tag{8}$$

where $W_{\text{e}}(\mathbf{y}_0, \mathbf{y}_1)$ and $L_{\text{e}}(\mathbf{y}_n, \mathbf{y}_{n-1})$ are the camera importance and the light radiance; $C(\bar{\mathbf{z}})$ is the product of a contribution at each vertex along the path:

$$C(\bar{\mathbf{z}}) = \prod_{i=1}^{n-1} \rho(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1}), \quad \text{and} \quad G(\bar{\mathbf{z}}) = \prod_{i=0}^{n-1} G(\mathbf{y}_i, \mathbf{y}_{i+1}) \tag{9}$$

is the product of a contribution at each segment along the path. We accommodate both surface reflection and subsurface transmission vertices by defining the vertex contribution as

$$\rho(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1}) = \begin{cases} \rho_{\text{r}}(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1}), & \mathbf{y}_i \text{ is surface refl.} \\ S_w(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1}), & \mathbf{y}_i \text{ is subsurface trans.} \end{cases} \tag{10}$$

and we accommodate both vacuum and subsurface segments by defining the segment contribution as

$$G(\mathbf{y}_i, \mathbf{y}_{i+1}) = \begin{cases} R_d(\mathbf{y}_i, \mathbf{y}_{i+1}), & \mathbf{y}_i\mathbf{y}_{i+1} \text{ is subsurface} \\ V(\mathbf{y}_i, \mathbf{y}_{i+1}) T(\mathbf{y}_i, \mathbf{y}_{i+1}), & \mathbf{y}_i\mathbf{y}_{i+1} \text{ is vacuum} \end{cases} \tag{11}$$

where

$$V(\mathbf{y}_i, \mathbf{y}_{i+1}) = \begin{cases} 1, & \text{no surface between } \mathbf{y}_i\mathbf{y}_{i+1} \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

$$T(\mathbf{y}_i, \mathbf{y}_{i+1}) = \frac{|\cos(\theta_i)|\,|\cos(\theta_{i+1})|}{\|\mathbf{y}_{i+1} - \mathbf{y}_i\|^2}, \tag{13}$$

and θ_i and θ_{i+1} are the angles between the surface normals at $\mathbf{y}_i$ and $\mathbf{y}_{i+1}$ and the segment joining them.

4.2 Differentiating the Path Integral with BSSRDF

The gradient of the path integral (7) with respect to scene parameters π is not necessarily the integral of the gradient of the path contribution (8), both because the path contribution with respect to geometric parameters like mesh vertex positions can be discontinuous due to the visibility changes in (12) and because the integration domain itself also depends on π. Zhang et al. [2021b] took the surface analog of the generalized Reynolds transport theorem [Cermelli et al. 2005], which computes the derivatives of an integral on a deforming domain, and defined parameter dependent functions on a fixed domain, making it possible to differentiate the path integral using the standard transport theorem with a non-deforming domain Γ and its boundary $\partial\Gamma$:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Gamma} \Phi(t) \mathrm{d}V = \int_{\Gamma} \frac{\partial \Phi(t)}{\partial t} \mathrm{d}V + \int_{\partial\Gamma} \Delta\Phi(t) \mathbf{v} \cdot \mathbf{n} \mathrm{d}A. \tag{14}$$

They refer to this parameter independent domain as material space. We will first review this material space and then apply it to our problem. Material space $\hat{M}$ is independent of π but connected to the moving surface M by a deformation χ that does depend on π:

$$\mathbf{p} = \chi(\mathbf{y}, \pi) \tag{15}$$

maps a surface vertex $\mathbf{y}$ to a point $\mathbf{p}$ in material space; $\bar{\mathbf{p}} = \mathbf{p}_0 \dots \mathbf{p}_n$ is a complete light path in $\hat{\Omega}_n \subset \hat{\Omega}$, where $\hat{\Omega}_n$ is the set of all the paths of length n in material space and $\hat{\Omega} = \cup_{n=1}^{\infty} \hat{\Omega}_n$ is the material path space. The measure on this domain is defined as the

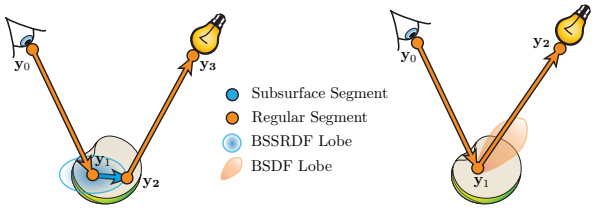


Figure 2: Our reconstruction process considers the path of length three (left) with a subsurface segment in the middle, and length two (right) for the specular BRDF lobe.

area product in material space $\mu(\overline{\mathbf{p}}) = \prod_{i=1}^{n} \mathrm{d}A(\mathbf{p}_i)$, and the path integral on material space becomes

$$I = \int_{\hat{\Omega}} \hat{f}(\overline{\mathbf{p}}) \mathrm{d}\mu(\overline{\mathbf{p}}), \tag{16}$$

where

$$\hat{f}(\overline{\mathbf{p}}) = \hat{W}_{\mathrm{e}}(\mathbf{p}_0, \mathbf{p}_1) C(\overline{\mathbf{p}}) G(\overline{\mathbf{p}}) \hat{L}_{\mathrm{e}}(\mathbf{p}_n, \mathbf{p}_{n-1}); \tag{17}$$

$C(\overline{\mathbf{p}}) = \prod_{i=1}^{n-1} \hat{\rho}(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})$ and $G(\overline{\mathbf{p}}) = \prod_{i=0}^{n-1} \hat{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ are the products of all scattering and geometry terms, in which

$$\hat{\rho}(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1}) = \rho(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1}) J(\mathbf{p}_i) \tag{18}$$

$$\hat{G}(\mathbf{p}_i, \mathbf{p}_{i+1}) = G(\mathbf{y}_i, \mathbf{y}_{i+1}), \tag{19}$$

and $J(\mathbf{p}) = \frac{|dA(\mathbf{y})|}{|dA(\mathbf{p})|}$ is the Jacobian of the map $\chi()$ between Ω and $\hat{\Omega}$.

The boundary $\partial\hat{\Omega}$ in this space is defined as the set of all paths that have at least one vacuum segment $\mathbf{y}_k\mathbf{y}_{k+1}$ on a visibility boundary: $\partial\hat{\Omega} = \cup_{n=1}^{\infty} \cup_{k=0}^{n-1} \partial\hat{\Omega}_{n,k}$, where $\partial\hat{\Omega}_{n,k}$ is the set of paths of length n with the k-th segment on the visibility boundary. Then the measure on the domain $\partial\hat{\Omega}_{n,k}$ is $\mu'_{n,k}(\overline{\mathbf{p}}) = dl(\mathbf{p}_k) \prod_{i\neq k}^{n} dA(\mathbf{p}_i)$.

In this fixed domain, we can apply the Reynolds transport theorem Eq. (14) to write the gradients of the path integral in material space Eq. (16) as the sum of an interior part that integrates the gradient of Eq. (17) and a boundary part that integrates the change of path contribution along the discontinuity boundary:

$$\frac{\partial I}{\partial \pi} = \frac{\mathrm{d}}{\mathrm{d}\pi} \int_{\hat{\Omega}} \hat{f}(\overline{\mathbf{p}}) \mathrm{d}\mu(\overline{\mathbf{p}}) = \underbrace{\int_{\hat{\Omega}} \frac{\partial(\hat{f}(\overline{\mathbf{p}}))}{\partial \pi} \mathrm{d}\mu(\overline{\mathbf{p}})}_{\text{interior}} + \underbrace{\int_{\partial\hat{\Omega}} \Delta\hat{f}(\overline{\mathbf{p}}) \mathbf{v}_B \mathrm{d}\mu'(\overline{\mathbf{p}})}_{\text{boundary}}. \tag{20}$$

where $\mathbf{v}_B$ is the normal velocity with respect to scene parameters of the path point on the boundary edge and the $\Delta\hat{f}(\overline{\mathbf{p}})$ is the change of path contribution across the boundary.

4.3 Differentiable Rendering for BSSRDF Capture

For our capture setup, shown in Fig. 2, which always has one point light, one pinhole camera, and a subsurface material object with indirect illumination neglected, we specialize Eq. (16) to

$$I = \int_{\hat{\Omega}_3} \hat{f}(\overline{\mathbf{p}}) \mathrm{d}\mu(\overline{\mathbf{p}}) + \int_{\hat{\Omega}_2} \hat{f}(\overline{\mathbf{p}}) \mathrm{d}\mu(\overline{\mathbf{p}}) \tag{21}$$

$$= \underbrace{\int_{\hat{M}} \int_{\hat{M}} \hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3) \mathrm{d}A(\mathbf{p}_1) \mathrm{d}A(\mathbf{p}_2)}_{I_3 \text{ with BSSRDF}} + \underbrace{\int_{\hat{M}} \hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2) \mathrm{d}A(\mathbf{p}_1)}_{I_2 \text{ with BRDF}}, \tag{22}$$

where the contribution $\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3)$ from a path with a subsurface segment in the middle is

$$W_{\mathrm{e}} V(\mathbf{y}_0\mathbf{y}_1) S_w(\mathbf{y}_1) J(\mathbf{p}_1) R_d(\mathbf{y}_1, \mathbf{y}_2) S_w(\mathbf{y}_2) J(\mathbf{p}_2) V(\mathbf{y}_2\mathbf{y}_3) L_{\mathrm{e}}, \tag{23}$$

and similarly, the contribution from the path consisting of two vacuum segments is $\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2) = W_{\mathrm{e}} V(\mathbf{y}_0\mathbf{y}_1) J(\mathbf{y}_1) \rho_{\mathrm{r}}(\mathbf{y}_0\mathbf{y}_1\mathbf{y}_2) V(\mathbf{y}_1\mathbf{y}_2) L_{\mathrm{e}}$.

Then differentiating the BSSRDF part of Eq. (22) with respect to scene parameters π using Eq. (20) gives us

$$\partial I_3/\partial\pi = \frac{\mathrm{d}}{\mathrm{d}\pi} \int_{\hat{\Omega}_3} \hat{f}(\overline{\mathbf{p}}) \mathrm{d}\mu(\overline{\mathbf{p}}) \tag{24}$$

$$= \underbrace{\int_{\hat{M}} \int_{\hat{M}} \partial\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3)/\partial\pi \mathrm{d}A(\mathbf{p}_1) \mathrm{d}A(\mathbf{p}_2)}_{\text{interior}} \tag{25}$$

$$+ \underbrace{\int\int \Delta\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3) \mathbf{v}_B(\mathbf{p}_2) \mathrm{d}A(\mathbf{p}_1) \mathrm{d}l(\mathbf{p}_2)}_{\text{boundary on } \mathbf{p}_2\mathbf{p}_3} \tag{26}$$

$$+ \underbrace{\int\int \Delta\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3) \mathbf{v}_B(\mathbf{p}_1) \mathrm{d}l(\mathbf{p}_1) \mathrm{d}A(\mathbf{p}_2)}_{\text{boundary on } \mathbf{p}_0\mathbf{p}_1}, \tag{27}$$

where $\mathbf{v}_B(\mathbf{p})$ is the velocity (gradient w.r.t π) of the visibility boundary in the direction of the normal to $dl(\mathbf{p})$, shown in Fig. 3.

4.4 Monte Carlo Estimation of the Derivatives

The path integral and the differentiable path integral are both estimated by a Monte Carlo algorithm: $\langle I \rangle = \frac{1}{N} \sum^{N} \frac{\hat{f}(\overline{\mathbf{x}})}{p(\overline{\mathbf{x}})}$; and in our problem setup, particularly,

$$\langle I_3 \rangle = \frac{1}{N} \sum^{N} \frac{\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3)}{p(\mathbf{y}_1)p(\mathbf{y}_2)}, \quad \langle I_2 \rangle = \frac{1}{N} \sum^{N} \frac{\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2)}{p(\mathbf{y}_1)}, \tag{28}$$

where the path with three segments contains one subsurface segment in the middle. Then the interior term in Eq. (25) is estimated by

$$\frac{1}{N} \sum^{N} \frac{\partial\hat{f}(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3)/\partial\pi}{p(\mathbf{y}_1)p(\mathbf{y}_2)}, \tag{29}$$

where the path is sampled just as in conventional path tracing.

To estimate the integrals in Eq. (27) and Eq. (26), paths are sampled on the discontinuity boundaries, including the primary visibility boundary (27) which relates to silhouette edges in the image, and the secondary visibility boundary (26), which relates to shadow edges cast on the mesh surface. Unlike previous work with opaque surface materials, where the discontinuity caused by the shadow edge is directly projected back to the camera, in our case this discontinuity is blurred by the BSSRDF on the surface and its motion indirectly affects the outgoing radiance at surface points near the shadow edge.

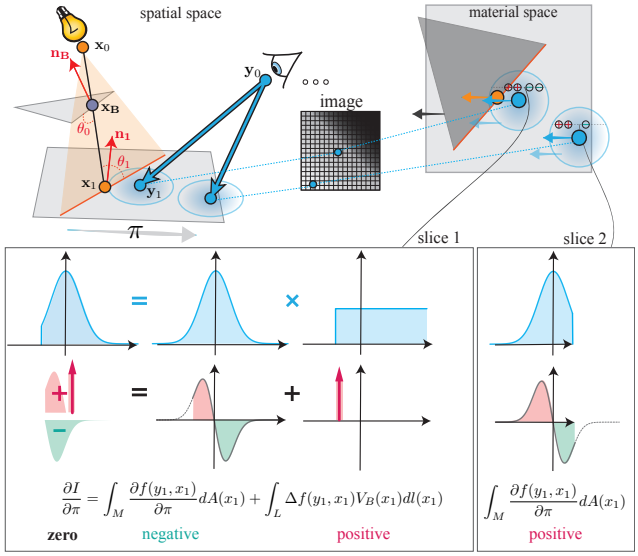


Figure 3: Illustration of interior and boundary contributions to the derivative. A fixed point light $\mathbf{x}_0$ and triangular occluder cast a shadow on a homogeneous translucent square which is moving rightward within its plane. Away from the square's boundary, the blurred shadow in the image is unaffected by this motion. A ray from the camera $\mathbf{y}_0$ hits the plane at $\mathbf{y}_1$ which is close to the shadow edge and far from the boundary. This path has zero derivative but neither (25) **nor** (26) **is zero. Rather the interior contribution is negative and the boundary contribution is positive, and they balance to zero.**

Fig. 3 shows an example of gradient estimation with a secondary visibility boundary (see caption for details). We use this example to validate our gradient estimations in Sec. 6.

5 OPTIMIZER

5.1 Variance Analysis of Loss Function

SGD requires an unbiased estimate of the gradient of the loss function $g(\pi)$. It is tempting to apply the loss to an unbiased estimate of $\mathbf{I}(\pi)$ at each iteration, then compute its derivative, but this produces biased results.

For instance, if we use the L_2 difference between an unbiased rendering $\tilde{\mathbf{I}}(\pi)$ and $\mathbf{I}_{\text{ref}}$ as the loss, SGD will minimize the expected value of this quantity, which is

$$\langle(\tilde{\mathbf{I}}(\pi) - \mathbf{I}_{\text{ref}})^2\rangle = \langle\tilde{\mathbf{I}}(\pi)^2\rangle - 2\langle\tilde{\mathbf{I}}(\pi)\rangle\mathbf{I}_{\text{ref}} + \mathbf{I}_{\text{ref}}^2. \tag{30}$$

The loss we really want is the difference between the exact rendering $\mathbf{I}(\pi)$, which is the expected value of $\tilde{\mathbf{I}}(\pi)$, and the reference:

$$(\langle\tilde{\mathbf{I}}(\pi)\rangle - \mathbf{I}_{\text{ref}})^2 = \langle\tilde{\mathbf{I}}(\pi)\rangle^2 - 2\langle\tilde{\mathbf{I}}(\pi)\rangle\mathbf{I}_{\text{ref}} + \mathbf{I}_{\text{ref}}^2. \tag{31}$$

The difference (30) − (31) is exactly the variance of $\tilde{\mathbf{I}}(\pi)$, and for noisy estimates this error dominates (30), biasing the solution towards parameters that produce lower variance. One example of this problem could be found in Fig. 8 (d) where using L2 loss at low sample rate per iteration converges to a wrong value. This problem can be avoided by taking care to compute an unbiased estimate of the loss derivative. One approach is to unwrap the loss function and back propagate the gradient to the parameters using $\frac{\partial g(I(\pi))}{\partial I(\pi)}\frac{\partial I(\pi)}{\partial \pi}$ [Nimier-David et al. 2020], which requires computing independent estimates for $\partial g/\partial \mathbf{I}$ and $\partial \mathbf{I}/\partial \pi$ by computing the image and its derivative in two passes. An alternative, used in this work, is to define the loss itself using two independent renderings as in Eq. (6). This approach is slightly more efficient because it effectively computes a two-sample estimate of the loss with two differential renderings, rather than a one-sample estimate with one rendering and one differential rendering.

5.2 Large Step Optimization

Mesh-based geometry optimization can be challenging due to triangle self intersections during surface evolution. Previous methods either rely on additional Laplacian smoothing losses [Ravi et al. 2020] or specific geometry processing software [Brochu et al. 2009] to avoid self intersections, but this can be costly in optimization. Recently, Nicolet et al. [2021] introduced a new parameterization of vertex positions and second-order preconditioned gradient descent, so called *large steps*, demonstrating superior performance in reconstructing both geometry and texture. We implemented this algorithm for optimizing geometry and textures in our pipeline. total

Surface evolution. Starting from a sphere or a roughly aligned mesh, we downsample the geometry to a low-polygon initial mesh. During the optimization we apply a remeshing step [Botsch and Kobbelt 2004] periodically to produce a coarse-to-fine sequence of meshes. Combined with this isotropic remeshing, we apply the preconditioned gradients (via the *large steps* algorithm) on vertex positions to evolve the surface and refine geometric details.

Material optimization. We apply a total variation smoothing prior on top of *large steps* for material texture optimization, defined as $\mathcal{L}_{\text{tv}}(\alpha) := \lambda_{\text{tv}} \sum_{i,j}(|\alpha[i+1,j]-\alpha[i,j]|+|\alpha[i,j+1]-\alpha[i,j]|)$, where $\alpha[i,j]$ indicates the value of the (i,j)-th pixel in the texture map.

6 IMPLEMENTATION DETAILS AND EXPERIMENTS

Differentiable Renderer. We implemented our inverse hetergeneous BSSRDF model based on the open-source *PSDR-CUDA* [Zhao and Yan 2021] differentiable renderer, leveraging GPU raytracing and automatic differentiation as well as its reference implementation for opaque surface reconstruction for comparison. For importance sampling the BSSRDF over surfaces, we follow the importance sampling strategies in [King et al. 2013]. To estimate the gradient contribution from secondary discontinuity boundaries, we sample points on shadow edges. This is done by sampling points on the triangle edges, connecting them with the point light, and tracing rays back against the surface; the resulting intersection points are right on the shadow edges.

Validation of autodiff derivatives. In Fig. 4, we validate the derivatives calculated via automatic differentiation in our path integral framework with BSSRDF.

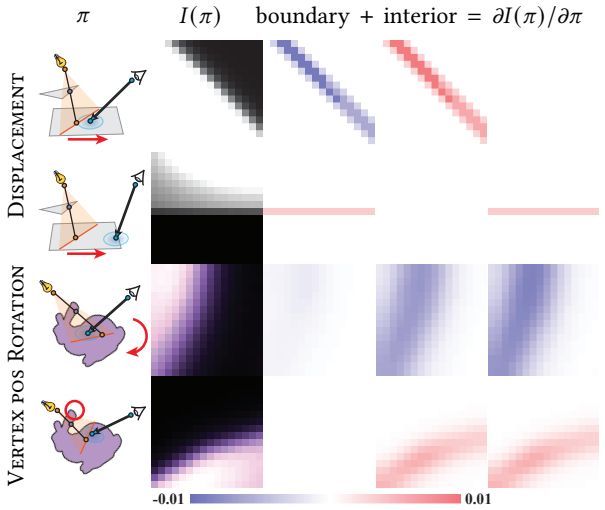


Figure 4: Validation of differentiable path integral framework with BSSRDF. In each row we show a scene configuration, the rendered image, our interior and boundary derivatives, which sum up to match the finite difference (last column). The changing scene parameters is marked by a red arrow (translation, rotation) or circle (vertex position).

Reconstructing with opaque BSDF vs. translucent BSSRDF. We compare geometry reconstruction quality with different material assumptions. As shown in Fig. 5, opaque assumption breaks when target object is more translucent, while our method handles both cases robustly due to the subsurface scattering material assumption, yielding more faithful shading and reconstruction of shape.

Reconstructing with volumetric path tracing vs. BSSRDF. In Fig. 7 we show an equal-sample comparison of reconstruction quality between differentiable volumetric path tracing [Zhang et al. 2021b] and our method. The bunny scene with a rough dielectric surface and highly scattering subsurface comes from [Zhang et al. 2021b]. The reference is rendered with volumetric path tracing using Mitsuba [Jakob 2010]. We initialize the optimization with the same learning rate and optimize until the loss converges or diverges. We experimented on 32spp and 128spp per optimization iteration for both methods. The previous work [Zhang et al. 2021b] failed to converge at such a low sample rate while ours stably converges to reference values.

Reconstruction of real-world translucent objects. We further examine the effectiveness of our approach on real data. As shown in Fig. 6, we captured 51 multi-view photographs of a bar of soap, which exhibits visibly translucent appearance under point-light illumination. We start from a perfect cube mesh, and initialize the single scattering albedo texture and spatially-varying surface roughness map from constant values, both at resolution 512×512. We optimize a uniform extinction coefficient for the soap and dragon-fruit, and a 512x512 extinction texture for the kiwi. Besides the image loss, we also use the total variation loss to ensure smoothness of the texture and a silhouette loss to ensure convergence of the bottom part of the shape. Our end-to-end differentiable rendering framework jointly recovers their shapes and heterogeneous translucent materials. We show global illumination of these results in Fig. 1.

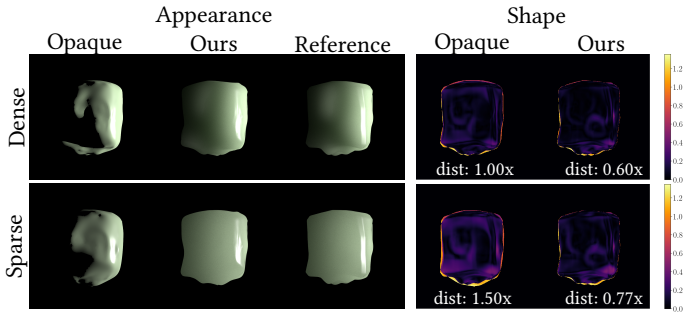


Figure 5: Comparison between an opaque BRDF (baseline) and a BSSRDF (ours) fit to synthetic images. Initialized from a gray sphere, we jointly optimize the shape and homogeneous translucent materials. We ran the optimization at the same learning rate and equal number of iterations for both methods. With the opaque BRDF model, the reconstruction quality of the geometry decreases when the target object is more translucent (i.e., Dense vs. Sparse), while our method robustly reconstructs the shape and translucent materials in both cases. We use Hausdorff distance to measure the difference between reconstructed shape and the reference shape.

7 DISCUSSION & CONCLUSION

We have presented a technique for jointly recovering heterogeneous subsurface scattering materials and geometry of real-world translucent objects and shown that a diffusion-based BSSRDF representation is more efficient and robust than volumetric path tracing for this problem. Compared to prior methods that require expensive capture systems and specific lighting patterns, our work advances the practical goal of capturing of translucent objects, and our methods can enable low-cost handheld acquisition of objects that exhibit translucency as an important aspect of their appearance.

However, there are still limitations. First, our method fails to reconstruct some high-frequency specular details from real-world data. This is due to the limited and fixed mesh resolution that our current implementation can afford. A possible future direction is to optimize a normal or displacement map along with other textures. Second, our current BSSRDF representation is a one-layer model, which we observed to have difficulties in reconstructing fruits with seeds beneath the surface where parallax presents a challenge. This indicates the necessity of a multi-layer subsurface model for handling these more complex objects. Finally, extending the framework to support capturing under environment lighting would be beneficial for more practical consumer-level applications.

ACKNOWLEDGMENTS

We would like to thank Cheng Zhang and Zihan Yu for discussions about inverse volumetric rendering and for their help in running their code. We thank Mengqi (Mandy) Xia for her help in capturing the real data. The bunny model was provided by the Stanford 3D Scanning Repository. This work was supported by National Science Foundation Grants 1704540 and 1900783 and a gift from Adobe.

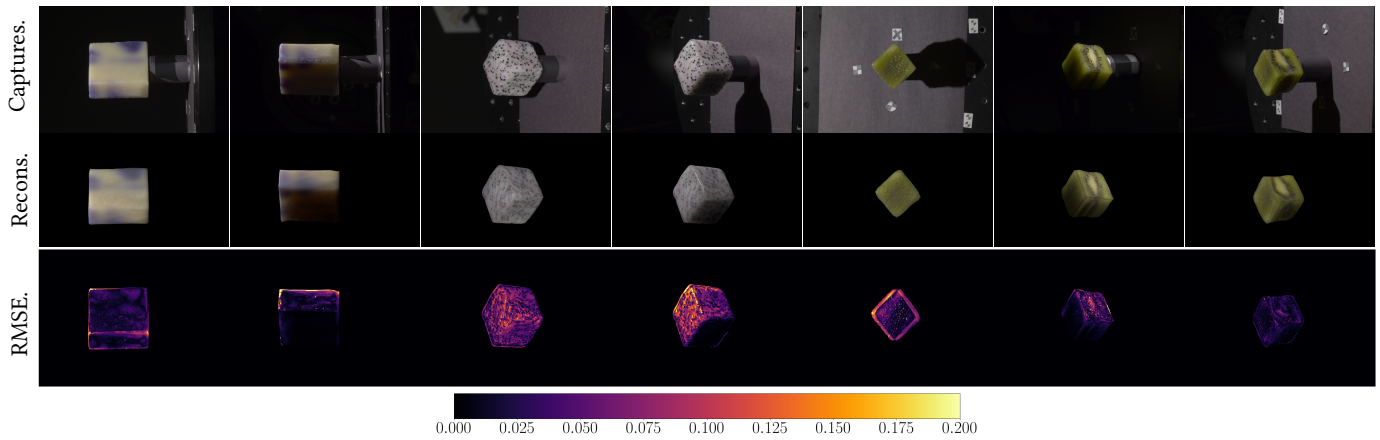
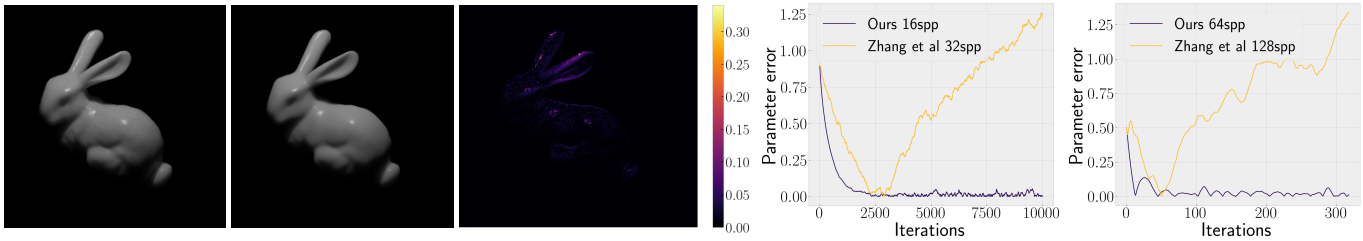
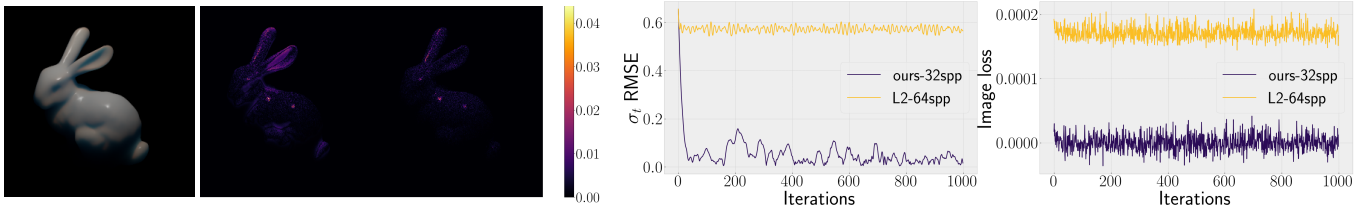


Figure 6: We demonstrate our end-to-end reconstruction framework on real data (column 1-2: bar of soap; column 3-4: cube cut from dragon fruit; column 5-7: cube cut from kiwi fruit) which are captured using a calibrated camera with a point light. Fifty-one views are used. We jointly recover the shape, from a cube, and the material parameters including the scattering albedo texture, surface roughness texture, both at resolution 512×512, and a uniform single scattering extinction coefficient (soap and dragon fruit) or a scattering extinction coefficient texture (for kiwi fruit).



(a) Reference (b) Our BSSRDF (c) Error map (d) 32-spp comparison (e) 128-spp comparison

Figure 7: Equal-sample comparison on inverse rendering of a translucent bunny (a) by optimizing using our BSSRDF model vs. brute-force volumetric path tracing [Zhang et al. 2021b]. We show the (b) rendered image with converged parameters that are optimized with our BSSRDF model and (c) its corresponding error map w.r.t. the ground truth. The differences around the ears are mainly due to global illumination that is ignored in our method. In (d) and (e), we visualize the parameter error during the inverse rendering process between ours and Zhang et al. [2021b] at 32spp and 128spp, respectively. Note that with our BSSRDF representation the optimization converges stably towards the target value while [Zhang et al. 2021b] diverges due to the noise of gradient estimation and bias in loss function. We see it diverging here at sample counts where ours works, but note that as demonstrated in their paper, it does converge at around 1k samples per pixel.



(a) Reference (b) L2 loss (c) Our loss (d) Parameter RMSE (e) Image Loss

Figure 8: Equal-sample comparison of the L2 loss vs. our proposed dual-buffer loss. The reference image (a) is a path-traced translucent bunny, and we run optimizations using both losses to recover the RGB attenuation coefficient. At each iteration we render a 64spp image for the L2 loss and two 32spp buffers for our loss. We fix the BSSRDF importance sampling distribution for all iterations, making it easier to analyze the variance of the L2 loss (this doesn't change the overall result of the comparison). In (b) and (c), we show the residual error images using the converged parameters from each optimization. As shown in (d) and (e), optimizing with our loss (purple) converges to match the reference better both in parameter error and image loss, compared to using L2 loss (yellow) which converges to a biased value

REFERENCES

Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. 2019. Inverse path tracing for joint material and lighting estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020).

Mario Botsch and Leif Kobbelt. 2004. A remeshing approach to multiresolution modeling. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 185–192.

Tyson Brochu et al. 2009. El Topo: Robust Topological Operations for Dynamic Explicit Surfaces. https://github.com/tysonbrochu/eltopo.

Paolo Cermelli, Eliot Fried, and Morton E. Gurtin. 2005. Transport relations for surface integrals arising in the formulation of balance laws for evolving fluid interfaces. *Journal of Fluid Mechanics* 544 (Dec. 2005), 339–351. https://doi.org/10.1017/S0022112005006695

Subrahmanyan Chandrasekhar. 1960. *Radiative transfer*. Courier Corporation.

Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Towards learning-based inverse subsurface scattering. In *The IEEE International Conference on Computational Photography (ICCP)*.

Yanyun Chen, Xin Tong, Jiaping Wang, Stephen Lin, Baining Guo, and Heung-Yeung Shum. 2004. Shell texture functions. *ACM Trans. Graph.* 23, 3 (2004).

Eugene d'Eon and Geoffrey Irving. 2011. A quantized-diffusion model for rendering translucent materials. *ACM Trans. Graph.* 30, 4 (2011).

Bo Dong, Kathleen D Moore, Weiyi Zhang, and Pieter Peers. 2014. Scattering parameters and surface normals from homogeneous translucent materials using photometric stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Craig Donner, Jason Lawrence, Ravi Ramamoorthi, Toshiya Hachisuka, Henrik Wann Jensen, and Shree Nayar. 2009. An empirical BSSRDF model. In *ACM SIGGRAPH 2009 papers*. 1–10.

Craig Donner, Tim Weyrich, Eugene d'Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz. 2008. A layered, heterogeneous reflectance model for acquiring and rendering human skin. *ACM Trans. Graph.* 27, 5 (2008).

Oskar Elek, Denis Sumin, Ran Zhang, Tim Weyrich, Karol Myszkowski, Bernd Bickel, Alexander Wilkie, and Jaroslav Krivanek. 2017. Scattering-aware texture reproduction for 3D printing. *ACM Trans. Graph.* 36, 6 (2017).

Jeppe Revall Frisvad, Toshiya Hachisuka, and Thomas Kim Kjeldsen. 2014. Directional dipole model for subsurface scattering. *ACM Trans. Graph.* 34, 1 (2014).

Christian Fuchs, Tongbo Chen, Michael Goesele, Holger Theisel, and Hans-Peter Seidel. 2007. Density estimation for dynamic volumes. *Computers & Graphics* 31, 2 (2007).

Christian Fuchs, Michael Goesele, Tongbo Chen, and Hans-Peter Seidel. 2005. An empirical model for heterogeneous translucent objects. *ACM SIGGRAPH Sketches* (2005).

Abhijeet Ghosh, Tim Hawkins, Pieter Peers, Sune Frederiksen, and Paul Debevec. 2008. Practical modeling and acquisition of layered facial reflectance. ACM New York, NY, USA, 1–10.

Ioannis Gkioulekas, Anat Levin, and Todd Zickler. 2016. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision (ECCV)*.

Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. 2013. Inverse volume rendering with material dictionaries. *ACM Trans. Graph.* 32, 6 (2013).

Michael Goesele, Hendrik PA Lensch, Jochen Lang, Christian Fuchs, and Hans-Peter Seidel. 2004. Disco: acquisition of translucent objects. *ACM Trans. Graph.* 23, 3 (2004).

Ralf Habel, Per H Christensen, and Wojciech Jarosz. 2013. Photon beam diffusion: A hybrid monte carlo method for subsurface scattering. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library.

Tim Hawkins, Per Einarsson, and Paul Debevec. 2005. Acquisition of time-varying participating media. *ACM Trans. Graph.* 24, 3 (2005).

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. 2001. A Practical Model for Subsurface Light Transport. *ACM Trans. Graph.* (2001).

Pramook Khungurn, Daniel Schroeder, Shuang Zhao, Kavita Bala, and Steve Marschner. 2015. Matching Real Fabrics with Micro-Appearance Models. *ACM Trans. Graph.* 35, 1 (2015).

Alan King, Christopher Kulla, Alejandro Conty, and Marcos Fajardo. 2013. BSSRDF Importance Sampling. In *ACM SIGGRAPH 2013 Talks* (Anaheim, California) *(SIGGRAPH '13)*. Association for Computing Machinery, New York, NY, USA, Article 48, 1 pages. https://doi.org/10.1145/2504459.2504520

Aviad Levis, Yoav Y Schechner, Amit Aides, and Anthony B Davis. 2015. Airborne three-dimensional cloud tomography. In *IEEE International Conference on Computer Vision (ICCV)*.

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6 (2018).

Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.* 38, 6 (2019).

Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. 2021. Unified Shape and SVBRDF Recovery using Differentiable Monte Carlo Rendering. *Computer Graphics Forum* 40, 4 (2021).

Yasuhiro Mukaigawa, Kazuya Suzuki, and Yasushi Yagi. 2009. Analysis of subsurface scattering based on dipole approximation. *IPSJ TCVA* 1 (2009).

Adolfo Munoz, Jose I Echevarria, Francisco J Seron, Jorge Lopez-Moreno, Mashhuda Glencross, and Diego Gutierrez. 2011. BSSRDF estimation from single images. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library.

Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. 2018. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Trans. Graph.* 37, 6 (2018), 1–12.

Srinivasa G. Narasimhan, Mohit Gupta, Craig Donner, Ravi Ramamoorthi, Shree K. Nayar, and Henrik Wann Jensen. 2006. Acquiring Scattering Properties of Participating Media by Dilution. *ACM Trans. Graph.* 25, 3 (2006).

F.E. Nicodemus, J.C. Richmond, J.J. Hsia, I.W. Ginsberg, and T. Limperis. 1977. *Geometrical Considerations and Nomenclature for Reflectance*. National Bureau of Standards.

Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large steps in inverse rendering of geometry. *ACM Trans. Graph.* 40, 6 (2021).

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020).

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. Graph.* 38, 6 (2019).

Thomas Klaus Nindel, Tomáš Iser, Tobias Rittig, Alexander Wilkie, and Jaroslav Křivánek. 2021. A gradient-based framework for 3D print appearance optimization. *ACM Trans. Graph.* 40, 4 (2021), 1–15.

Jan Novák, Iliyan Georgiev, Johannes Hanika, Jaroslav Krivánek, and Wojciech Jarosz. 2018. Monte Carlo methods for physically based volume rendering.. In *SIGGRAPH Courses*.

Marios Papas, Christian Regg, Wojciech Jarosz, Bernd Bickel, Philip Jackson, Wojciech Matusik, Steve Marschner, and Markus Gross. 2013. Fabricating translucent materials using continuous pigment mixtures. *ACM Trans. Graph.* 32, 4 (2013).

Pieter Peers, Karl Vom Berge, Wojciech Matusik, Ravi Ramamoorthi, Jason Lawrence, Szymon Rusinkiewicz, and Philip Dutré. 2006. A compact factored representation of heterogeneous subsurface scattering. *ACM Trans. Graph.* 25, 3 (2006).

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. 2020. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501* (2020).

Tobias Rittig, Denis Sumin, Vahid Babaei, Piotr Didyk, Alexey Voloboy, Alexander Wilkie, Bernd Bickel, Karol Myszkowski, Tim Weyrich, and Jaroslav Křivánek. 2021. Neural Acceleration of Scattering-Aware Color 3D Printing. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 205–219.

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ying Song, Xin Tong, Fabio Pellacini, and Pieter Peers. 2009. Subedit: a representation for editing measured heterogeneous subsurface scattering. *ACM Trans. Graph.* 28, 3 (2009).

Denis Sumin, Tobias Rittig, Vahid Babaei, Thomas Nindel, Alexander Wilkie, Piotr Didyk, Bernd Bickel, J KŘivánek, Karol Myszkowski, and Tim Weyrich. 2019. Geometry-aware scattering compensation for 3D printing. *ACM Trans. Graph.* 38, 4 (2019).

Sarah Tariq, Andrew Gardner, Ignacio Llamas, Andrew Jones, Paul Debevec, and Greg Turk. 2006. Efficient estimation of spatially varying subsurface scattering parameters. *Vision, Modeling, and Visualization (VMV2006)* (2006), 129–136.

Xin Tong, Jiaping Wang, Stephen Lin, Baining Guo, and Heung-Yeung Shum. 2005. Modeling and rendering of quasi-homogeneous materials. *ACM Trans. Graph.* 24, 3 (2005).

Eric Veach. 1998. *Robust Monte Carlo methods for light transport simulation*. Stanford University.

Zdravko Velinov, Marios Papas, Derek Bradley, Paulo Gotardo, Parsa Mirdehghan, Steve Marschner, Jan Novák, and Thabo Beeler. 2018. Appearance capture and modeling of human teeth. *ACM Trans. Graph.* 37, 6 (2018).

Delio Vicini, Vladlen Koltun, and Wenzel Jakob. 2019. A learned shape-adaptive subsurface scattering model. *ACM Trans. Graph.* 38, 4 (2019).

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path replay backpropagation: differentiating light paths using constant memory and linear time. *ACM Trans. Graph.* 40, 4 (2021).

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. *Rendering techniques* 2007 (2007), 18th.

Jiaping Wang, Shuang Zhao, Xin Tong, Stephen Lin, Zhouchen Lin, Yue Dong, Baining Guo, and Heung-Yeung Shum. 2008. Modeling and Rendering of Heterogeneous Translucent Materials Using the Diffusion Equation. *ACM Trans. Graph.* 27, 1 (2008).

Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, et al. 2006.

Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph.* 25, 3 (2006).

Lifan Wu, Guangyan Cai, Ravi Ramamoorthi, and Shuang Zhao. 2021. Differentiable time-gated rendering. *ACM Trans. Graph.* 40, 6 (2021), 1–16.

Shinyoung Yi, Donggun Kim, Kiseok Choi, Adrian Jarabo, Diego Gutierrez, and Min H Kim. 2021. Differentiable transient rendering. *ACM Trans. Graph.* 40, 6 (2021), 1–11.

Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo estimators for differential light transport. *ACM Trans. Graph.* 40, 4 (2021).

Cheng Zhang, Zhao Dong, Michael Doggett, and Shuang Zhao. 2021a. Antithetic sampling for Monte Carlo differentiable rendering. *ACM Trans. Graph.* 40, 4 (2021).

Cheng Zhang, Bailey Miller, Kan Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020).

Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A differential theory of radiative transfer. *ACM Trans. Graph.* 38, 6 (2019).

Cheng Zhang, Zihan Yu, and Shuang Zhao. 2021b. Path-Space Differentiable Rendering of Participating Media. *ACM Trans. Graph.* 40, 4 (2021).

Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016. Downsampling scattering parameters for rendering anisotropic media. *ACM Trans. Graph.* 35, 6 (2016).

Shuang Zhao and Kai Yan. 2021. Path-Space Differentiable Renderer. https://psdr-cuda.readthedocs.io/en/latest/.