Reconstructing Translucent Thin Objects from Photos

XI DENG, Cornell University, USA
LIFAN WU, NVIDIA, USA
BRUCE WALTER, Cornell University, USA
RAVI RAMAMOORTHI, NVIDIA & University of California, San Diego, USA
EUGENE D'EON, NVIDIA, New Zealand
STEVE MARSCHNER, Cornell University, USA
ANDREA WEIDLICH, NVIDIA, Canada

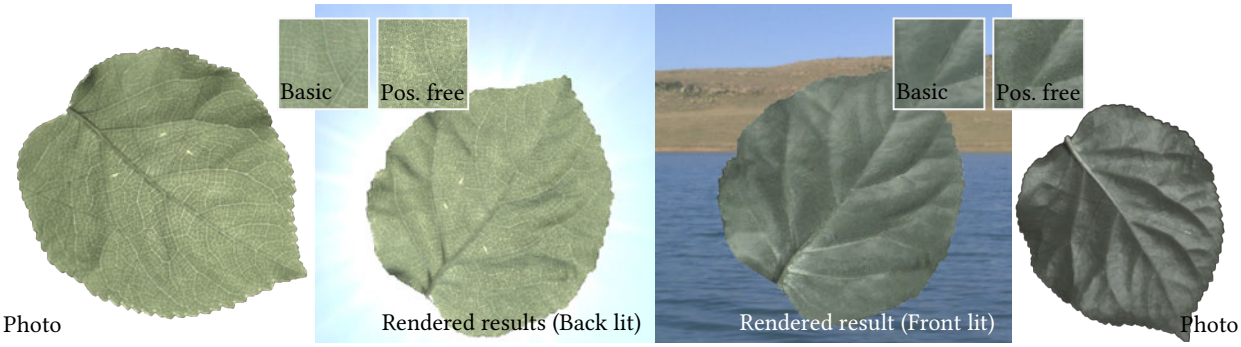


Fig. 1. Rendered results of our reconstruction pipeline for thin translucent materials. We jointly reconstruct geometry and materials through differentiable rendering to obtain spatially varying optical properties for a position-free layered volumetric model to simulate the appearance of objects like paper or leaves.

The joint reconstruction of shape and appearance for translucent objects from real-world data poses a challenge in computer graphics, especially when dealing with complex layered materials like leaves or paper. The traditional assumption of diffuse transmittance falls short, and more accurate Monte-Carlo-based models are often needed to reproduce their appearance. To accurately capture the translucent appearance, an acquisition system needs to be carefully designed. Additionally, there are three challenges for inverse rendering: First, a large number of unknown parameters make the optimization problem difficult. Second, the Monte Carlo (MC) renderer introduces noise, which the optimization is sensitive to, especially when dealing with complex material models such as rough dielectric surfaces and highly scattering participating media. Last, MC estimators using long light paths (up to 32 bounces in our case) create a large computation graph in memory, making the gradient back-propagation costly. To address those challenges, we present an affordable and fast acquisition pipeline that can capture spatially-varying reflectance and transmission at the same time, using a two-phase optimization. We first initialize the geometry with the traditional vision method and then fit a simple and fast appearance model. Thereafter, we use the estimated parameters to initialize a second optimization using a more expensive volumetric model, which converges faster and more reliably from this favorable starting position. We also introduce a way to analyze each parameter's sensitivity to the noise in the measurements, which can be used in optimally selecting useful measurements for optimization. Furthermore, instead of iterating on the camera system, we also introduce a weighted ℓ_2 loss as an alternative for selecting useful pixels from existing measurements.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: differentiable rendering, inverse rendering, 3D reconstruction

ACM Reference Format:
Xi Deng, Lifan Wu, Bruce Walter, Ravi Ramamoorthi, Eugene d'Eon, Steve Marschner, and Andrea Weidlich. 2024. Reconstructing Translucent Thin Objects from Photos. 1, 1 (October 2024), 11 pages. https://doi.org/10.1145/3680528.3687572

Authors' addresses: Xi Deng, xd93@cornell.edu, Cornell University, USA; Lifan Wu, lifanw@nvidia.com, NVIDIA, USA; Bruce Walter, bruce.walter@cornell.edu, Cornell University, USA; Ravi Ramamoorthi, ravir@cs.ucsd.edu, NVIDIA & University of California, San Diego, USA; Eugene d'Eon, edeon@nvidia.com, NVIDIA, New Zealand; Steve Marschner, srm@cs.cornell.edu, Cornell University, USA; Andrea Weidlich, aweidlich@nvidia.com, NVIDIA, Canada.

XXXX-XXXX/2024/10-ART $15.00
https://doi.org/10.1145/3680528.3687572

1 INTRODUCTION

Translucent thin objects with multi-layer structure, such as plant leaves and papers, are seen everywhere in our daily life; however, it remains difficult to model these objects' appearance with sufficient detail for high-quality rendering. This group of materials exhibits non-diffuse transmittance due to multiple scattering between layers which gives them a striking appearance when back-lit, and they can look markedly different between the two sides when front-lit.

Appearance capture research has produced many good methods for measuring opaque surfaces and fitting spatially-varying

reflectance models. Extending this work to thin translucent objects requires a model that encompasses both front and back reflectance and non-diffuse transmittance, along with measurement and fitting methods to determine the parameters.

In this work, we propose a new appearance capture method for translucent thin objects. A key principle of our work is to use a unified physics-based model that determines reflectance and transmittance under illumination and viewing from both sides, all from one set of parameters. Specifically, we adopt a model using thin volumetric layers separated by rough interfaces [Jakob et al. 2014], and compute the appearance using Monte Carlo methods [Guo et al. 2018]. As opposed to using separate models for the two sides' reflectances and for transmittance, a single model holds the promise of generalizing better from fewer observations, because it inherently produces a globally physically consistent appearance. We also propose a simple capture device to rapidly measure translucent surfaces, and a multi-phase inverse rendering pipeline to recover the parameters for rendering.

Using inverse rendering optimization to reproduce the spatially varying translucent appearance poses several challenges. First, the number of unknown geometry and material parameters is huge. Without a proper initialization scheme, the optimization cannot converge stably, so we use a two-phase optimization: first we optimize a simple surface BSDF model, and then we optimize a position-free layered volumetric model, initializing based on the result of the first step. The second challenge is that the optimization is sensitive to Monte Carlo noise from the differentiable renderer. To address that, we exploit sensitivity analysis to design a robust loss function for faster and more robust convergence. Finally, differentiable rendering involving long light paths is computationally expensive. To accelerate rendering and optimization, we introduce a new pre-integration technique to accelerate the computation of reflectance or transmittance from the physical parameters.

We demonstrate our system by measuring a number of plant leaves and art papers, materials with interesting spatially varying appearances. The raw measurements from our device will be released as a dataset to be used in material appearance research.

All together, the main contributions of our work include:

- A simple and fast acquisition system that captures images of reflected and transmitted light from several viewpoints under a number of lighting conditions, which is tailored to rapidly provide sufficient data to fit our models.
- A new pipeline to jointly reconstruct both shape and physics-based appearance parameters using differentiable rendering.
- An improvement to the position-free layered material model by pre-integrating one dimension of distance sampling.
- A robust loss function for the appearance optimization, selected based on sensitivity analysis.
- A dataset including measurements and reconstructions of leaves and paper materials.

These contributions advance the field of inverse rendering by showing how to apply the new tools of differentiable physics based rendering to a practical problem in appearance modeling. Our work shows that while inverse rendering is a flexible and powerful approach, care in the choice of model and optimization strategy is required to actually achieve the results this paradigm is capable of.

2 RELATED WORK

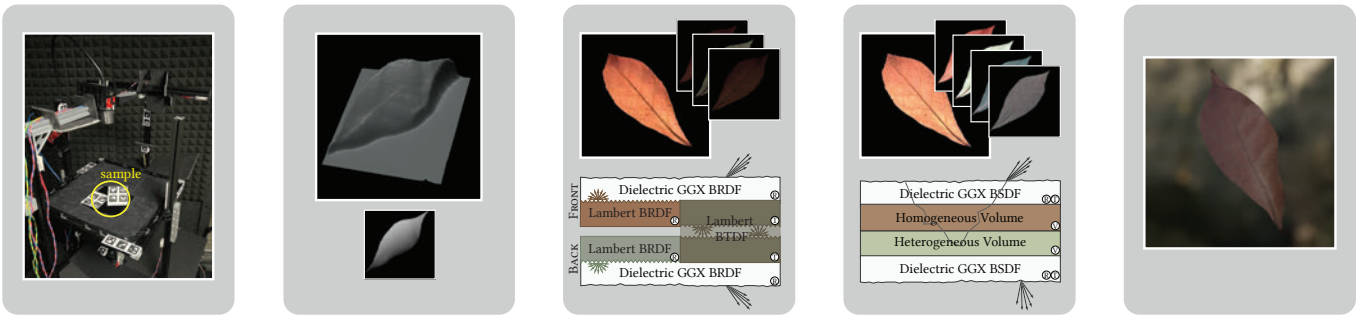
Appearance capture. There have been lots of efforts in capturing the optical properties of real-world objects including leaves [Bousquet et al. 2005], paper [Papas et al. 2014], wood [Marschner et al. 2005], human hair [Marschner et al. 2003], animal fur [Yan et al. 2015], etc. These works usually rely on sophisticated acquisition systems built with complex physical devices with moving parts, such as spherical gantries or gonio-photometers [Dupuy and Jakob 2018; Guarnera et al. 2016; Matusik et al. 2003]. In contrast, our acquisition device is quite simple, with only fixed lights and cameras and no moving parts. Some SVBRDF capture systems [Aittala et al. 2015; Ma et al. 2023; MÃijller et al. 2005; Nam et al. 2018] also use fixed or uncalibrated cameras and lights, e.g. by using a light dome or a hand-held camera with built-in flash. Like our work this avoids the need for moving parts, but these works focus on standard surface reflectance models and don't support transmission. To handle transmission, we put cameras and lights on both sides of the object during measurement. Unlike existing work that measures translucency [Gardner et al. 2003; Merzbach et al. 2019], we measure both directional and spatial variation in transmittance.

Modeling translucent materials. Subsurface scattering in translucent materials is governed by the radiative transfer equation [Chandrasekar 1960]. However, simulating full volumetric light transport in 3D volumes is computationally expensive [Novák et al. 2018] and is unsuitable for our use. We show that a thin layered (plane-parallel) approximation is expressive enough to model the appearance of thin translucent objects such as leaves and papers, but simple and fast enough to use in inverse rendering.

Researchers have proposed various physically based layered material models [Belcour 2018; Guo et al. 2018; Hanrahan and Krueger 1993; Jakob 2015; Jakob et al. 2014; Weidlich and Wilkie 2007; Zeltner and Jakob 2018] in past decades. Among those layered models, we choose to use the stochastic ones [Guo et al. 2018; Xia et al. 2020] because they do not rely on empirical approximations or on precomputations that limit parameter variation. As a result, our reconstructed layered model can reproduce the real captured appearance accurately, without the concern of quality loss due to the use of a restricted or less expressive model.

Simulating multiple scattering in layered materials is easier than in 3D geometry but still requires a Monte Carlo integration with many samples. For more efficiency, recent learning-based layered models [Fan et al. 2022; Guo et al. 2023; Wang et al. 2022] use neural networks to approximate the multiple-scattering BSDFs of layered materials. Using such approximations to accelerate our rendering and optimization is possible, but we decided to keep our approach physically-based to avoid errors in generalization.

Physically based differentiable rendering. Physically based differentiable rendering is the key component of optimization-based inverse rendering, which has been used to solve for the optical



(a) Photographic capture (b) Initial geometry reconst. (c) Basic model (d) Layered model (e) Rendering

Fig. 2. Overview of our acquisition pipeline. We (??) first capture the object with three cameras and eleven lights before we reconstruct the geometry (??). We use a simplified model to capture the principal appearance of the object (??) with differentiable rendering, convert the data, and feed the output to a volumetric layered model (??) which can better capture the shape of the transmission lobe. The resulting material model can be used in a classic path tracer (??).

parameters of a wide range of materials such as translucent volumes [Gkioulekas et al. 2013; Hašan and Ramamoorthi 2013] and cloth [Khungurn et al. 2016; Zhao et al. 2016]. Most of those works assume the scene geometry to be static to avoid discontinuities caused by visibility. Recent advances in differentiable rendering algorithms [Bangaru et al. 2020; Jiang et al. 2020; Li et al. 2018; Loubet et al. 2019; Zhang et al. 2020, 2019] addressed this problem and properly handled the discontinuities, allowing joint optimization of shapes and materials [Deng et al. 2022; Luan et al. 2021; Xing et al. 2023, 2022; Yan et al. 2023]. Our inverse rendering pipeline is based on the path-space differentiable rendering formulation [Zhang et al. 2020] and jointly optimizes the shape and material properties of translucent thin objects.

Several works have applied variance reduction techniques such as control variates [Nicolet et al. 2023], antithetic sampling [Yu et al. 2022; Zhang et al. 2021], and resampled importance sampling [Chang et al. 2023; Wang et al. 2023] to the often noisy derivative estimators. We also propose a way to reduce variance in our path integral by substituting an analytic solution for some sampled dimensions, which is orthogonal to these variance reduction methods.

3 OVERVIEW

Our reconstruction pipeline aims to jointly infer the geometry and appearance of a translucent thin object. It consists of four stages: photo acquisition, geometry reconstruction, simple material reconstruction, and position-free material model reconstruction. The input for our pipeline is a thin translucent material sample such as paper, a leaf, a petal, etc. The desired output is a model for both the shape and the optical scattering properties of the object.

Acquisition System. Our apparatus, detailed in Sec. 4, consists of three cameras and eleven light sources fixed in space on both sides of the sample. The lights are switched on in succession to capture HDR images for all light/camera pairs in a few minutes. This speed is important because thin biological samples like leaves would dry and change shape and appearance during a longer measurement.

Material Model. Objects like leaves and paper are geometrically thin, but thick in an optical sense—light bounces multiple times inside the material before it finally exits, and negligible light passes directly through. Their properties vary across the surface, but slowly compared to the thickness, so we adopt the *position-free* approach [Guo et al. 2018] and assume that reflection and transmission at any point on the surface are defined by light scattering in a homogeneous layered medium [Jakob et al. 2014].

Depending on the type of object being captured, various layer configurations could be used, but for all the examples in our paper we used a stack of two volume layers, with rough dielectric interfaces on the top and bottom. To account for the surface shape, we use a displacement map. Thus in total our model is fully described by five textures—height, front and back albedo, front and back optical thickness—and five global parameters: front and back surface roughness, the mean cosine of the phase function per layer, and a single index of refraction. Our optimization also uses a simplified model to locate a starting point, which replaces the volume layers with separate diffuse reflectances for front and back, plus a diffuse transmittance. Details of the two models and our methods for rendering them are in Sec. 5.

Optimization. Unlike most 2D appearance capture systems, we adopt a full inverse rendering approach for fitting the model parameters to our observations. This is necessary because the model parameters define the surface appearance via a Monte Carlo rendering process, and once we adopt this approach it gives us the freedom to use any type of lighting and to optimize the surface shape jointly with the optical properties. By giving it the right input, we enable the optimizer to automatically employ strategies that encompass stereo triangulation and reflectance and scattering property estimation to best explain the observations.

Previous work in differentiable rendering in principle solves the problem of appearance fitting. When reconstructing a few appearance parameters in synthetic scenarios [Li et al. 2018; Loubet et al. 2019; Zhang et al. 2020] or simpler material models from real data [Deng et al. 2022; Luan et al. 2021], standard stochastic optimizers have been shown to find solutions when initialized randomly. However, for our problem, the optimization converges poorly because of the ill-conditioned nature of the problem. Similarity theory [Zhao et al. 2014] shows there are approximate null directions in

the parameter space, along which changes to the parameters have minimal effect on the external appearance. The slow convergence is compounded by the expense of computing derivatives of renderings with long scattering paths, even using state-of-the-art differentiation frameworks on powerful GPUs.

Sec. 6 details how we greatly improve the convergence of the optimization by first solving a sequence of easier fitting problems to obtain a good starting point for optimizing the full model, and Sec. 5.4 introduces a new pre-integration technique to improve the per-iteration performance.

4 ACQUISITION

Our acquisition system, which can be seen in Fig. 2(a) is meant to rapidly capture enough image data to sufficiently constrain the parameters of our appearance model. This requires observations of both sides of the surface under illumination from both sides, with enough viewing and illumination directions to observe the 3D shape and the non-diffuse reflection and transmission. In this section we describe the design considerations and summarize the procedures for calibration and data processing.

4.1 System Design

In our system the measurement sample must be held in place while being illuminated and photographed from both sides. The samples we measure are typically flexible and lightweight; this makes the system prone to problems with vibration, but also makes it unnecessary to have a very rigid support for the sample.

Considerations of vibration and cost/complexity led us to a design with no moving parts, but instead a set of fixed cameras and lights. With no vibration in the system, we decided not to use a rigid support such as a sheet of glass, which would cause problems with refraction and unwanted reflections. Instead the object is held in place using a sparse grid made of thin fishing line. Pixels that are affected by the lines could be masked out in the optimization loss and filled in from other cameras, but in this paper we have simply used the complete images directly.

Number of lights and cameras. For each point on the surface, we need to infer displacement, per-layer volume parameters, and surface roughness on two sides; a fundamental design question is how many views and how many illumination conditions per view are needed. We use three cameras: two on the front, which provide the ability to triangulate the height and additional views of specular reflections (which are typically stronger on the front side of leaves), plus one on the back. One extra camera is placed on the front only for the geometry reconstruction purpose and collecting data for validation. Since the object is translucent, lighting it from either side gives useful data from cameras on both sides. We use a set of lights that are turned on one by one, so we obtain three RGB measurements per camera, per light.

Our system uses 11 lights, 6 on the front and 5 on the back, which are positioned to provide a range of illuminations including specular, near-retroreflection, and grazing angles. This provides in total 33 RGB measurements per surface position, which provides sufficient redundancy over the 13 spatially varying parameters of the two-layered model used in our results.

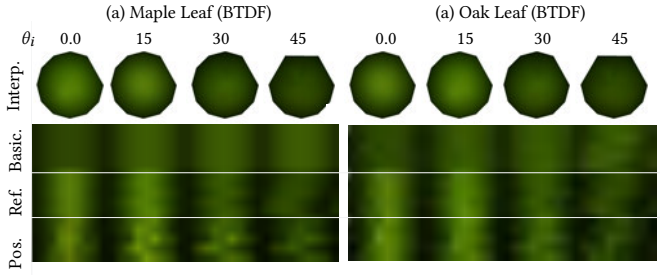


Fig. 3. BTDFs from leaf measurements [Roth 2020] (first and third rows) compared against fits with the basic and position-free appearance models (second and fourth rows). The data shows non-diffuse transmittance, and the diffuse transmittance assumption of the basic model is inadequate. The position-free model fits the data much better.

Hardware. We use 5MP RGB cameras from Allied Vision (Alvium 1800 U-500C), paired with Fujinon 9mm lenses providing a field of view around 30°. With this setup, we are able to completely measure samples with dimensions of up to 10cm × 11cm. The light sources are chip-on-board LED arrays, each acting as a 6.0mm diameter circular Lambertian source.

4.2 Image Processing

For each exposure we capture an exposure series, beginning with a 2s exposure time and successively halving the time until no pixels are saturated. Including waiting time for camera synchronization the full capture sequence takes approximately 25 minutes for 52 high dynamic range images. The cameras record 10-bit one-channel images in a Bayer color pattern; we use standard methods to merge the exposures into a floating-point high dynamic range (HDR) image, then demosaic the image in the HDR domain using the adaptive homogeneity-directed (AHD) algorithm [Hirakawa and Parks 2003].

The resulting image is in the color space of the camera sensor, and we apply white balancing and a color matrix transformation, using parameters determined during calibration, to transform into the linear sRGB color space.

5 APPEARANCE MODELS

5.1 Calibration

The calibration of our measurement setup involves determining the camera intrinsics, the 3D poses of all the cameras and sources relative to the object plane, the color characteristics of the cameras, and the intensities of the light sources.

Camera intrinsics calibration was carried out using the OpenCV camera calibration toolkit. The locations of the light sources and of a number of reference points within the apparatus were determined by photographing a collection of Aruco markers using a mobile phone camera and running a bundle adjustment using OpenCV. A subset of those markers visible to each camera was used to calibrate that camera's pose, and a subset of the markers located on the approximate object plane were used to establish coordinates centered on that plane. (The object plane does not need to be located exactly because it is only a reference plane for displacement and the object surface need not be exactly in this plane.)

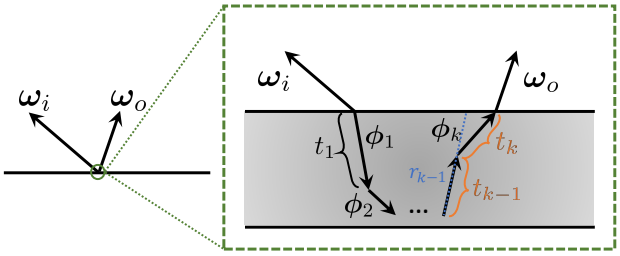


Fig. 4. Illustrating a path sample $\bar{\mathbf{z}}$ in the position-free layered BSDF formulation. The transmittances of t_{k-1} and t_k (shown in orange) are pre-integrated.

A gray card was photographed in several orientations so that it was seen by each camera and illuminated by each source. The pixel intensities in these images were used to establish per-camera and per-light scale factors that correct for camera sensitivity and light intensity so that pixel values can be interpreted as radiance measurements relative to a unit source intensity.

Photographs of the Macbeth Color Checker test chart were used to obtain color balancing factors and a camera-to-sRGB transformation matrix for each camera.

In this section, we will introduce two appearance models we use in the inverse rendering process: the basic two-sided SVBSDF model and the position-free layered model [Guo et al. 2018] (see Fig. 2 for a schematic illustration) and will discuss the parameter conversion between the two models for initializing the layered model. In addition, we propose a modification to the position-free layered model by pre-integraing one dimension of distance sampling, which can significantly improve the rendering efficiency.

5.2 Two-Sided SVBSDF Model

To represent the different appearances of an object when viewing it from opposite sides, our simple two-sided SVBSDF model (illustrated in Fig. 2(c)) consists of two opaque reflectance layers, and each reflectance layer has a Lambertian diffuse lobe and a glossy GGX lobe [Walter et al. 2007]. In addition, we add a Lambertian transmission lobe in the middle to handle translucency. Given a point $\mathbf{x}$, the incident direction $\boldsymbol{\omega}_i$, and the outgoing direction $\boldsymbol{\omega}_o$, our two-sided SVBSDF

$$F_s(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \begin{cases} w_{\mathrm{D}}^{\mathrm{front}} f_{\mathrm{D}}^{\mathrm{front}} + f_{\mathrm{GGX}}^{\mathrm{front}} + w_{\mathrm{T}} f_{\mathrm{T}}, & \cos\boldsymbol{\omega}_i > 0, \\ w_{\mathrm{D}}^{\mathrm{back}} f_{\mathrm{D}}^{\mathrm{back}} + f_{\mathrm{GGX}}^{\mathrm{back}} + w_{\mathrm{T}} f_{\mathrm{T}}, & \cos\boldsymbol{\omega}_i < 0, \end{cases} \tag{1}$$

where $f_{\mathrm{D}}^{\mathrm{front}}$, $f_{\mathrm{GGX}}^{\mathrm{front}}$, $f_{\mathrm{D}}^{\mathrm{back}}$, $f_{\mathrm{GGX}}^{\mathrm{back}}$, and f_{T} are SVBRDFs or SVBTDFs (their input parameters $\mathbf{x}$, $\boldsymbol{\omega}_i$, and $\boldsymbol{\omega}_o$ are omitted in the equation), and the accompanying BSDF lobe weights $w_{\mathrm{D}}^{\mathrm{front}}$, $w_{\mathrm{D}}^{\mathrm{back}}$ and w_{T} contain the residual energy of the GGX lobe on which we apply energy correction [Turquin 2019] to avoid darkening. A similar SVBSDF formulation has been used for modeling leaf appearance [Bousquet et al. 2005; Wang et al. 2005].

5.3 Position-Free Layered Model

The major weakness of our basic two-sided surface model is its crude approximation of translucency and multiple scattering. A more accurate appearance model is the physically based position-free layered model [Guo et al. 2018] because it simulates light transport within the object in an unbiased way. Its only assumption is that the light entrance and exiting locations are the same, which is approximately true when the surface is thin geometrically. Our position-free layered model consists of two dielectric interfaces on the front and the back side of a surface and two volumetric layers in between (shown in Fig. 2(d)). The dielectric layers have spatially homogeneous roughness values, and the volumetric layers have spatially varying single-scattering albedo c and extinction coefficients σ_t.

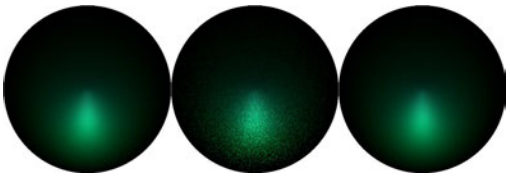


Fig. 5. Results of pre-integration for one dimension in single scattering. From left to right: Reference, position-free model and pre-integrated position-free model, showing the lower hemisphere for an incident direction in the upper hemisphere, both computed with 100 samples/pixel. Note the reduction in noise compared to the non-integrated version.

Given a point $\mathbf{x}$ and a pair of directions $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$, the layered BSDF $F_p(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ aggregates the light scattering inside layers, which can be formulated as an integral over light paths $\bar{\mathbf{z}}$:

$$F_p(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \int f(\bar{\mathbf{z}}) \mathrm{d}\mu(\bar{\mathbf{z}}), \tag{2}$$

where a path sample $\bar{\mathbf{z}}$ is defined by a sequence of distance t and directions $\boldsymbol{\phi}$, i.e., $\bar{\mathbf{z}} = (\boldsymbol{\omega}_i, \boldsymbol{\phi}_1, t_1, \ldots, \boldsymbol{\phi}_k, t_k, \boldsymbol{\omega}_o)$ as shown in Fig. 4, and f is the measurement contribution function:

$$f(\bar{\mathbf{z}}) = \rho(\boldsymbol{\omega}_i, \boldsymbol{\phi}_1)\rho(\boldsymbol{\phi}_k, \boldsymbol{\omega}_o)\mathrm{Tr}(t_k)\prod_{i=1}^{k-1}\rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_{i+1})\mathrm{Tr}(t_i). \tag{3}$$

The scattering function ρ is either a BSDF or a phase function, depending on the current depth inside the layers, and Tr is the transmittance function in volumetric rendering. For simplicity in this formulation, rays passing between the layers are handled with a null-scattering event represented by a delta phase function. The position-free layered BSDF F_p can be estimated using Monte Carlo integration, and we use the unidirectional path tracing to sample.

5.4 Pre-Integrating One Dimension of Distance

The noise introduced by the Monte Carlo estimation of the position-free layered model poses additional challenges in forward and inverse rendering. To reduce noise, we reformulate the integration and improve the estimation efficiency with negligible computational cost. The idea is to analytically pre-integrate the last distance sampling dimension along the ray, and then to replace the innermost integral over the two exponential terms in transmittance with a closed-form solution, which turns out to be another exponential term weighted by direction cosines (see Fig. 4).

By expanding Eq. (2) and rearranging the integrals, we have

$$F_p(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = I_D \int_0^{r_1} \cdots \int_0^{r_{k-1}} \prod_{i=1}^{k} \mathrm{Tr}(t_i) \mathrm{d}t_{k-1} \cdots \mathrm{d}t_1, \tag{4}$$

with $I_D = \int_S \cdots \int_S \rho(\boldsymbol{\omega}_i, \boldsymbol{\phi}_1)\rho(\boldsymbol{\phi}_k, \boldsymbol{\omega}_o) \prod_{i=1}^{k-1} \rho(\boldsymbol{\phi}_i, \boldsymbol{\phi}_{i+1}) \mathrm{d}\boldsymbol{\phi}_k \cdots \mathrm{d}\boldsymbol{\phi}_1$, S is the solid angle domain for directional samples and r_i is the distance between point $\mathbf{z}_i$ and the surface boundary along the direction $\boldsymbol{\phi}_i$. where $a = |\cos\theta|$ is the absolute cosine value between $\boldsymbol{\phi}_k$ and the surface normal. After rearranging Eq. (4), we get

$$\int_0^{r_1} \cdots \int_0^{r_{k-2}} \prod_{i=1}^{k-2} \mathrm{Tr}(t_i) I_{k-1} \mathrm{d}t_{k-2} \cdots \mathrm{d}t_1, \tag{5}$$

where $I_{k-1} = \int_0^{r_{k-1}} \mathrm{Tr}(t_{k-1})\mathrm{Tr}(t_k)\mathrm{d}t_{k-1}$ and it equals to

$$\begin{cases} \frac{\mathrm{Tr}(r_{k-1}/a_k)a_{k-1}}{\sigma_t(a_k - a_{k-1})}(1.0 - \mathrm{Tr}(\sigma_t/a_{k-1} - \sigma_t/a_k)), & \uparrow\uparrow, \\ \frac{\mathrm{Tr}(\zeta - r_{k-1}a_k/a_k)a_{k-1}}{\sigma_t(a_k + a_{k-1})}(1.0 - \mathrm{Tr}(\sigma_t/a_{k-1} + \sigma_t/a_k)), & \uparrow\downarrow, \end{cases} \tag{6}$$

where ζ is the depth of the layer. This will give us an integral with one fewer dimension:

$$F_p(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = I_D \int_0^{r_1} \cdots \int_0^{r_{k-2}} I_{k-1} \prod_{i=1}^{k-2} \mathrm{Tr}(t_i) \mathrm{d}t_{k-2} \cdots \mathrm{d}t_1$$

In theory, one can pre-integrate all the distance dimensions, and Bitterli and d'Eon [2022] have shown that the transmittance term can be estimated as a sum of exponential terms. However, this is limited to a single-layer configuration, and the performance degrades after ten bounces in our experiments. Since we want to keep the computation vectorizable for any path inside any layer for multiple bounces, as well as keeping the computational graph simple enough for differentiable rendering, our 1D pre-integration works best in our scenario.

5.5 Parameter Conversion

In our optimization we use the strategy of fitting a basic two-sided surface model and using the result to initialize the position-free layered model. These models use different input parameters: the basic one uses diffuse reflectance and diffuse transmittance whereas the position-free model needs c and σ_t as input. We achieve this by computing values of c and σ_t that will result in the diffuse reflectance computed in the first phase.

As can be seen in Fig. 6, directly interpreting the diffuse reflectance albedo as the volumetric single-scattering albedo leads to an appearance that depends on surface roughness, IOR, and phase function, and is generally a poor fit. Instead, we utilize an albedo mapping scheme [d'Eon 2022] that is commonly used in content production. Note that we do not need to convert the surface roughness or IOR of the GGX BRDF, because in this case the same model is shared by both phases.

6 OPTIMIZATION

Our inverse-rendering optimization relies on physically based differentiable rendering [Zhang et al. 2020] to estimate derivatives with respect to shape and material parameters. Due to the large number of unknown parameters, the ill conditioned nature of volume parameter inversion, and the Monte Carlo noise from derivative estimation, our optimization problem is quite prone to difficulties with convergence in practice, To make the optimization tractable, we reconstruct the shape and material of a thin translucent object in three steps. We first optimize the shape of the target object (Sec. 6)

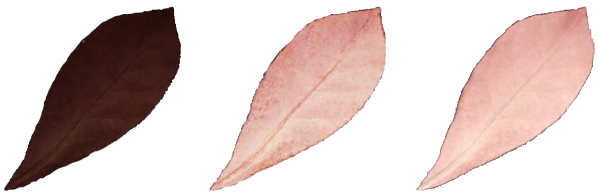


Fig. 6. Mapping reflectance to scattering albedo. The diffuse reflectance texture fit in the first reflectance optimization phase (left) contains information about the volumetric single-scattering albedo, but the reflectance values are far from the optimum scattering albedo (right). Our albedo mapping step computes a scattering albedo texture (center) that is much closer to the optimum, making it a good starting point for the final optimization phase.

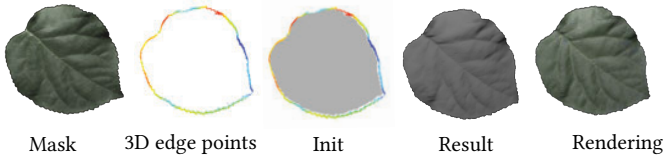


Fig. 7. Phases of geometry reconstruction. Left to right: one input image with its foreground mask; the 3D edge points colored by height; the smooth initial displacement map; the optimized displacement map; and the rendering matching the left image.

and fix it afterward. Then, we optimize the material parameters of our basic two-sided SVBSDF model. Finally, we solve for the volumetric material parameters of our position-free layered model, using the SVBSDF parameters for initialization (Sec. 5.5). We discuss the loss functions we use in Sec. 6.2.

6.1 Geometry Optimization

Correctly modeling the deviation of our object from a planar surface is important to avoid misregistration and inaccurate shading, as shown in Fig. 7. Since our objects are overall nearly planar, we take a 2.5D approach and use a displacement map to represent the object's surface, implemented by displacing the vertices of a regular mesh in the x-y plane only along the z-direction. This reduces the number of unknown geometry parameters compared to optimizing the 3D vertex positions, avoiding unwanted tangential motion and improving convergence.

In practice it is important to start the geometry optimization with a reasonable estimate of the geometry, particularly around the edges of the object. Since our input includes foreground masks that accurately locate the edges in image space, we use the three front views in a simple multi-view triangulation method to estimate the height at the edge. Specifically, we use edge detection on the masks to locate the edge, and then for each edge pixel in one view we search for edges along the epipolar lines in the other two views. Where all three images find edges consistent with a 3D point, we use the corresponding depth. We project the 3D boundary to the displacement base surface and solve a Laplace equation to find a smooth initial displacement map that is consistent with the reconstructed heights at the edge. We use this map to project the foreground masks and derive a consensus mask that is used to define the edges of the surface during optimization.

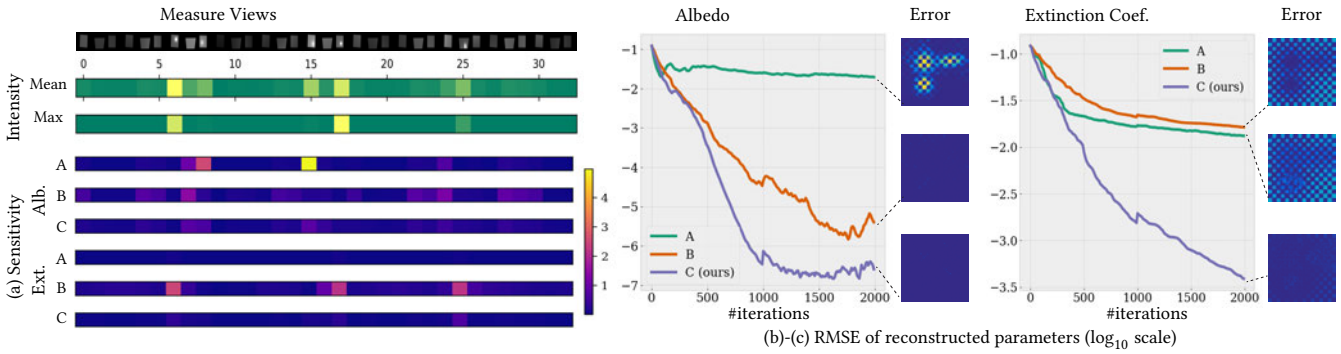


Fig. 8. Sensitivity and convergence analysis for fitting albedo and extinction coefficient textures. (a) We analyzed the sensitivity of the parameters to errors in the measurements using (A) ℓ_2 loss, (B) ℓ_2 loss over tone-mapped images, and (C) our variance-weighted ℓ_2 loss. With loss (A) the albedo is highly sensitive to views (7, 8, 15), and with loss (B) the extinction coefficient is highly sensitive to views (6, 17, 25). Loss (C) has low and balanced sensitivity to the inputs. We also show the mean and maximum intensity in each view, revealing that bright pixels are sometimes but not always associated with high sensitivity to noise. (b-c) We ran optimizations with the three losses, showing that losses (A) and (B) get stuck in local minima with high error, while loss (C) finds a good solution. The worst and best optimization performance are associated with the highest and lowest variation in sensitivity.

Starting from this initialization, we optimize the displacement map using the "large steps" preconditioner (Nicolet et al. [2021]). This phase produces as output a single one-channel displacement texture and coarse roughness and reflectance values for the basic model.

Our material optimization consists of two stages. For the first stage, we use our basic two-sided SVBSDF model. The goal is to find a set of surface reflectance and transmission parameters that result in a reasonable appearance match, so that they can be used for initializing the position-free layered model to further improve the reconstruction quality. During this stage, we keep the shape fixed and only optimize the material parameters, and all the training view-light pairs are used (except any that are held out as a test set as discussed in Sec. 7). This process converges quickly and produces texture maps for front and back diffuse reflectance and for diffuse transmittance, along with single parameters for front roughness, back roughness, and index of refraction. Then, we switch from the basic model to our position-free layered model by converting the reflectance maps to the spatially varying single-scattering albedo as described in Sec. 5.5.

6.2 Loss Functions

In constructing photometric loss functions for use with high dynamic range images, care is needed to balance the influence of bright and dark pixels. To study the choice of loss function, we performed a sensitivity analysis, shown in Fig. 8. By analyzing the loss derivative at the converged solution in a simple synthetic fitting problem with constant albedo and transmittance, we computed the sensitivity of each of these parameters on all the pixel values. We repeated this for each of three candidate losses: a standard ℓ_2 loss, an ℓ_2 loss computed on tonemapped images, and an ℓ_2 loss weighted by the inverse of the per-pixel variance of the rendered image. Fig. 8(a) shows that in both the ℓ_2 loss and in the tonemapped loss, the parameters are highly sensitive to pixel values in certain images, whereas with our weighted loss the sensitivity is well balanced. This advantage plays out in the optimization results shown in (b) and (c),

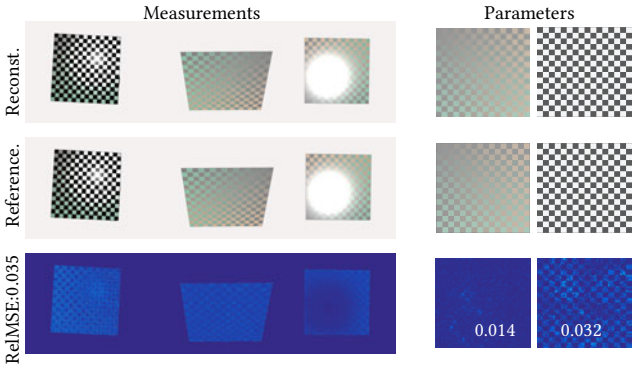


Fig. 9. We validate the optimization on synthetic data to reconstruct optical properties of the layers. The camera and light setup is the same as for the real system. In the third row, we color code the relative root mean square error between the reference and reconstructed data. Both measurements and parameter values are reconstructed with good quality.

where the optimization with our loss converges faster and to lower residuals than with the other losses. For these reasons we use the variance-weighted loss in all the results shown in this paper.

We also apply the total variation loss for smoother texture maps, as well as a range loss to limit the parameters to physically meaningful ranges. Note that at high roughness value (> 0.5) the GGX lobe has significant energy loss, which makes the energy correction mentioned in Sec. 5.2 necessary to prevent the optimization from over-fitting dark views.

7 EXPERIMENTS

We implement our differentiable rendering pipeline in the PSDR framework [Zhang et al. 2020] using Dr.Jit [Jakob et al. 2022]. The derivative of our appearance models (i.e., $\partial F_s/\partial \pi$ and $\partial F_p/\partial \pi$) can be computed using automatic differentiation. To capture multiple scattering, we set the maximum path length to 32 in all experiments. The PSDR method handles differentiating cast shadows, which are

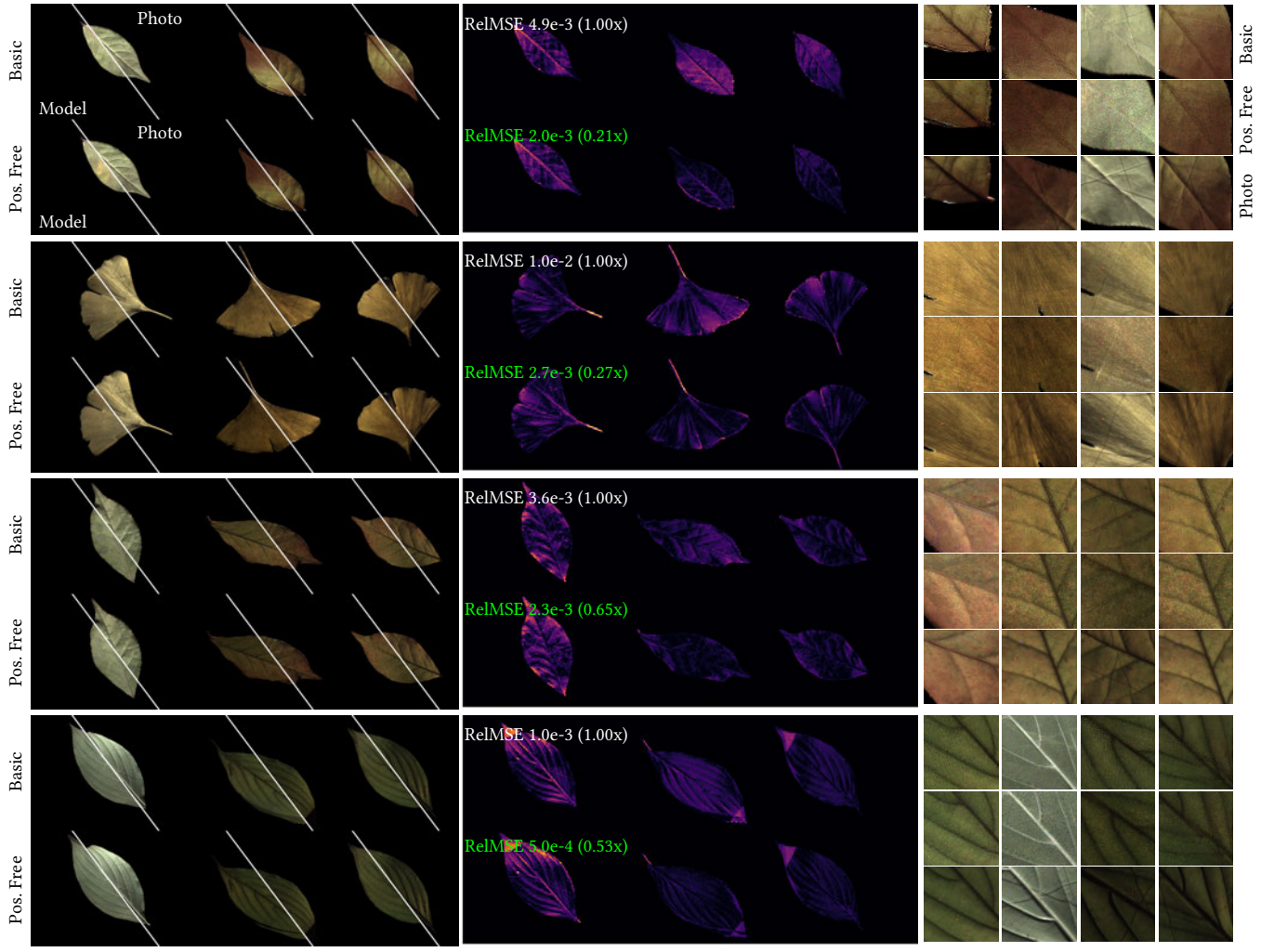


Fig. 10. **Validation** of reconstruction quality in **novel lighting**. Under new lighting, we show the root mean square error maps (RMSE) between the captured photo and the rendering of the reconstructed models and compare the reconstruction using a basic model and a position-free model initialized with the basic model. This validation experiment shows the quality of the reconstruction for both models. While the basic model can already capture the appearance quite well, the more accurate physically-based model has less energy mismatch in translucent energy numerically. Please note that for this experiment we omitted images we would normally use from the reconstruction process, i.e we have only 10 lights in this setup.

significant for grazing illumination when objects are significantly non-flat. We use the Adam optimizer [Kingma and Ba 2014] for inverse rendering. At each iteration, a view-light pair is randomly selected from the training set.

Capturing one sample takes around 25 minutes. The optimization for the geometry pass requires roughly an hour whereas the basic model takes about 40 minutes for 5000 iterations. Afterwards, the position-free model (128spp per iteration) runs for around 2 hours (2500 iterations again), using RTX 3090. All output textures have a resolution of 512x512 except the displacement map for which we use 256x256 size textures.

In Fig. 12, we show the validation of the geometry reconstruction on synthetic data where we reconstruct the geometry model from renderings of maple and Ginko leaves. The optimized parameters are displacement map and single value for reflectance and roughness. The "Init.Reconst." shows the result of the first phase of the geometry reconstruction and the "Opt.Reconst." shows the result of the gradient descent phase which refines the details in the interior. We use the Hausdorff distance to measure the difference between the ground truth and the reconstructions.

In Fig. 10 we validate our BSDF fits by holding out three images during the reconstruction process and testing the fitted model against these images, which show lighting directions that were not seen in the optimization. As can be seen, even our basic model is capable of reconstructing the appearance. However, the position-free model is consistently better at simulating the shape of the transmission lobes and is able to better predict these reference views.

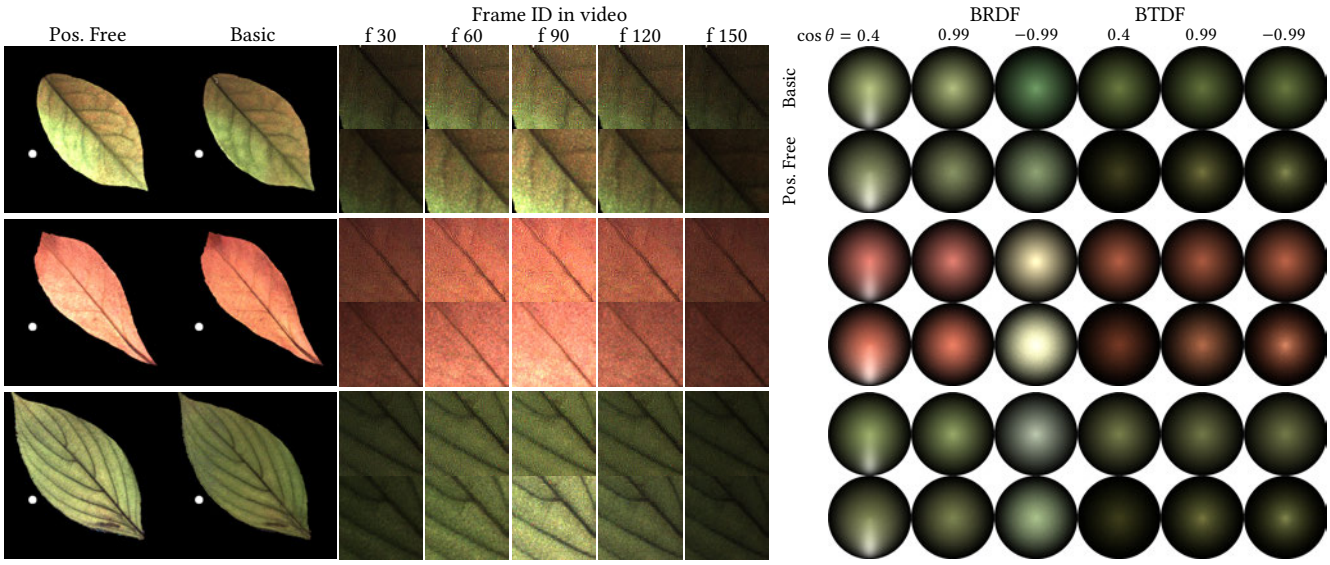


Fig. 11. We render the reconstructions with a light moving leftwards behind the objects (see attached videos). It can be seen that the transmission of the basic model is flatter than our position-free model. We also show the difference between two models by visualizing the BRDF and BTDF value at a randomly picked point on the object (with incident cosines equal to 0.4, 0.99, -0.99).

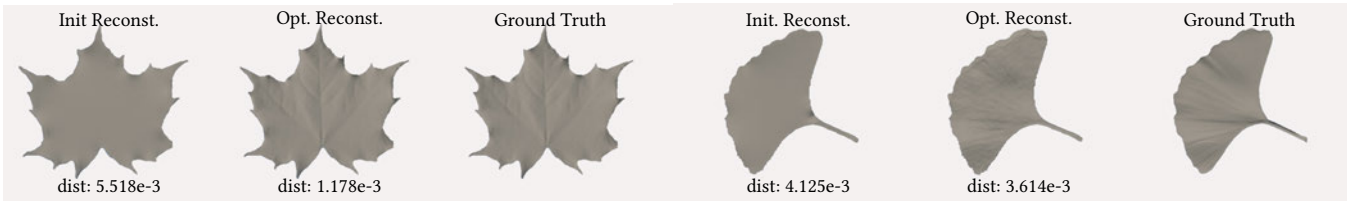


Fig. 12. We validate the geometry reconstruction on synthetic data. The camera and light setup is the same as for the real system. The first and the second columns are different phases of the geometry reconstruction which correspond to the 3rd and 4th column of Fig. 7 We use Hausdorff distance to measure the difference between the reconstructed shape and the reference shape. Both geometries are reconstructed with good quality.

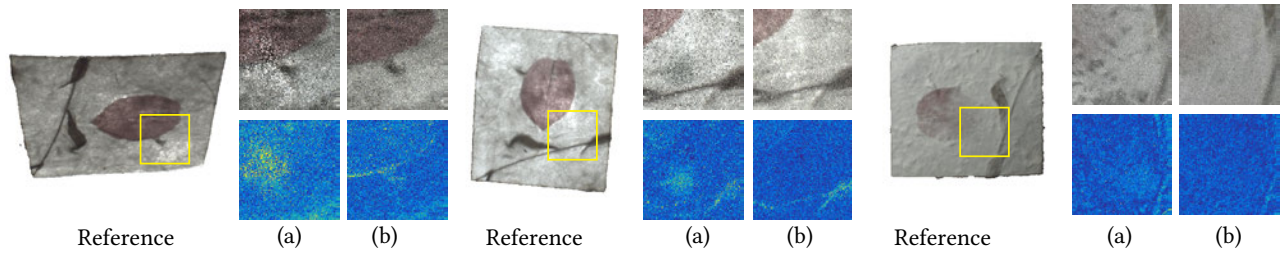


Fig. 13. We compare the reconstruction quality at the 1000th iteration between (a) random initialization and (b) initialization with the parameters of the basic model. With random initialization, all the parameters are optimized at the same time while our method allows us fix the parameters that are converted from the basic model, like roughness and single scattering albedo. The position-free model takes three times longer per iteration than the basic model.

A crucial requirement for successful optimization with the position-free model is to find good initialization values. Fig. 13 shows the outcome of initializing the position-free model with random numbers compared to initializing it with our basic model. The randomly initialized version struggles with the sharpness of the textures and sometimes fails to reconstruct the optical properties. Additionally, the randomly initialized version will need more iterations to achieve a similar quality, and it also takes approximately three times longer per iteration than the basic model.

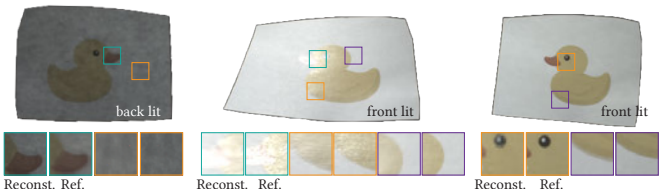


Fig. 14. We compare the renderings of models reconstructed from photos of a piece of printed paper against the photos. The renderings are rendered under three different viewing and lighting conditions, showing efficient reconstruction of transmittance, reflectance, and specular lobe. The parameters we reconstruct are the same as the leaves in Fig. 10 except for the roughness being a texture map.

In the supplementary video we rendered various objects lit from behind by moving lights. Since we use a physically based model, we avoid overfitting to certain directions and keep the appearance plausible when relighting. Stills from the video can be seen in Fig. 11.

8 CONCLUSION AND DISCUSSION

In this paper we presented a method for capturing both reflection and transmission from thin, translucent objects by physics-based inverse rendering, showing how differentiable rendering can be a tool for modeling the appearance of real materials in practice. Our work addresses *applied inverse rendering*, a topic that needs to be explored further to develop the techniques to set up and successfully solve large and complex inverse problems using differentiable renderers.

Currently we ignore the output of the transmission map reconstructed from the simple model, not yet converting it to initialize the parameters in the position-free model, but we think we could improve the initialization further by utilizing a two-stream model. We also do not reconstruct normal maps which would be able to simulate fine geometric differences on the front and back side, which a single displacement map cannot account for. We show in Fig. 14 renderings of reconstructed printed paper where the roughness varies spatially depending on where the printings are. Though spatially varying roughness is supported, a higher resolution of roughness map requires denser angular measurements and remains a good future extension of the system. One good future direction for performance improvement is to enable Path Replay Backpropagation [Vicini et al. 2021] inside the evaluation of the stochastic BSDF to allow for higher sample counts in inverse rendering.

ACKNOWLEDGMENTS

We are grateful to Aaron Lefohn for his support. This work starts with Xi's internship project at NVIDIA. We would like to thank Kai Yan and Zihan Yu for sharing their early version of PSDR-JIT code and Yu Guo for providing the Mitsuba version of position-free code for reference. We also would like to thank Beibei Xu, Yunchen Yu and Zhenli Wu for mailing leaves from the south to Ithaca in winter when there were no leaves. This work was supported in part by the Cornell Institute for Digital Agriculture, the National Science Foundation under award 2212084, and by a gift from NVIDIA Corporation.

REFERENCES

Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2015. Two-shot SVBRDF Capture for Stationary Materials. *ACM Trans. Graph.* 34, 4 (2015), 110:1–110:13.
Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased Warped-Area Sampling for Differentiable Rendering. *ACM Trans. Graph.* 39, 6 (2020), 245:1–245:18.
Laurent Belcour. 2018. Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators. *ACM Transactions on Graphics* 37, 4 (2018), 1.
Benedikt Bitterli and Eugene d'Eon. 2022. A PositionâĂŘFree Path Integral for Homogeneous Slabs and Multiple Scattering on Smith Microfacets. *Computer Graphics Forum* 41 (2022). https://api.semanticscholar.org/CorpusID:248495919
Laurent Bousquet, Stéphane Jacquemoud, and Ismael Moya. 2005. Leaf BRDF and BTDF measurements and model. In *ISPMSRS 2005 Conference Proceedings*. 459–461.
S. Chandrasekar. 1960. *Radiative Transfer*. Dover Publications.
Wesley Chang, Venkataram Sivaram, Derek Nowrouzezahrai, Toshiya Hachisuka, Ravi Ramamoorthi, and Tzu-Mao Li. 2023. Parameter-space ReSTIR for Differentiable and Inverse Rendering. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 18, 10 pages. https://doi.org/10.1145/3588432.3591512
Xi Deng, Fujun Luan, Bruce Walter, Kavita Bala, and Steve Marschner. 2022. Reconstructing Translucent Objects Using Differentiable Rendering. In *ACM SIGGRAPH 2022 Conference Proceedings*. Association for Computing Machinery, Article 38, 10 pages.
Eugene d'Eon. 2022. *A Hitchhiker's Guide to Multiple Scattering: Exact Analytic, Monte Carlo and Approximate Solutions in Transport Theory.*
Jonathan Dupuy and Wenzel Jakob. 2018. An adaptive parameterization for efficient material acquisition and rendering. *ACM Trans. Graph.* 37, 6, Article 274 (2018), 14 pages.
Jiahui Fan, Beibei Wang, Miloš Hašan, Jian Yang, and Ling-Qi Yan. 2022. Neural Layered BRDFs. In *Proceedings of SIGGRAPH 2022*.
Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. 2003. Linear light source reflectometry. 22, 3 (2003), 749âĂŞ758.
Ioannis Gkioulekas, Bei Xiao, Shuang Zhao, Edward H. Adelson, Todd Zickler, and Kavita Bala. 2013. Understanding the Role of Phase Function in Translucent Appearance. *ACM Transactions on Graphics* 32, 5 (Oct. 2013), 147:1–147:19. https://doi.org/10/gfzp5h
D. Guarnera, G.C. Guarnera, A. Ghosh, C. Denk, and M. Glencross. 2016. BRDF Representation and Acquisition. *Computer Graphics Forum* 35, 2 (2016), 625–650.
Jie Guo, Zeru Li, Xueyan He, Beibei Wang, Wenbin Li, Yanwen Guo, and Ling-Qi Yan. 2023. MetaLayer: A Meta-Learned BSDF Model for Layered Materials. *ACM Trans. Graph.* 42, 6, Article 222 (2023), 15 pages.
Yu Guo, Miloš Hašan, and Shaung Zhao. 2018. Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs. *ACM Trans. Graph.* 37, 6 (2018).
Pat Hanrahan and Wolfgang Krueger. 1993. Reflection from Layered Surfaces Due to Subsurface Scattering. New York, NY, USA, 165–174. https://doi.org/10/b4tw3j
Milovš Hašan and Ravi Ramamoorthi. 2013. Interactive Albedo Editing in Path-Traced Volumetric Materials. *ACM Transactions on Graphics* 32, 2 (April 2013), 11:1–11:11. https://doi.org/10/gfz5nk
K. Hirakawa and T.W. Parks. 2003. Adaptive homogeneity-directed demosaicing algorithm. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, Vol. 3. III–669. https://doi.org/10.1109/ICIP.2003.1247333
Wenzel Jakob. 2015. layerlab: A computational toolbox for layered materials. In *SIGGRAPH 2015 Courses (SIGGRAPH '15)*. ACM, New York, NY, USA. https://doi.org/10.1145/2776880.2787670
Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. 2014. A Comprehensive Framework for Rendering Layered Materials. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33, 4 (July 2014), 118:1–118:14. https://doi.org/10/f6cpsq
Wenzel Jakob, SÃŽbastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. Dr.Jit: A Just-In-Time Compiler for Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 41, 4 (July 2022). https://doi.org/10.1145/3528223.3530099
Y. Jiang, D. Ji, Z. Han, and M. Zwicker. 2020. SDFDiff: Differentiable Rendering of Signed Distance Fields for 3D Shape Optimization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 1248–1258. https://doi.org/10.1109/CVPR42600.2020.00133
Pramook Khungurn, Daniel Schroeder, Shuang Zhao, Kavita Bala, and Steve Marschner. 2016. Matching Real Fabrics with Micro-Appearance Models. 35, 1, Article 1 (dec 2016).
Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).
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), 222:1–222:11.
Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.* 38, 6 (2019), 228:1–228:14.
Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. 2021. Unified Shape and SVBRDF Recovery using Differentiable Monte Carlo Rendering. *Computer Graphics*

Forum 40 (2021).
Xiaohe Ma, Xianmin Xu, Leyao Zhang, Kun Zhou, and Hongzhi Wu. 2023. OpenSVBRDF: A Database of Measured Spatially-Varying Reflectance. *ACM Trans. Graph.* 42, 6, Article 254 (2023), 14 pages.
Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. 2003. Light Scattering from Human Hair Fibers. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 22, 3 (July 2003), 780. https://doi.org/10/b4cprd
Stephen R. Marschner, Stephen H. Westin, Adam Arbree, and Jonathan T. Moon. 2005. Measuring and modeling the appearance of finished wood. *ACM Trans. Graph.* 24, 3 (2005), 727âĂŞ734.
Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. 2003. A Data-Driven Reflectance Model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 22, 3 (July 2003), 759–769. https://doi.org/10/fjjgv8
S. Merzbach, M. Hermann, M. Rump, and R. Klein. 2019. Learned Fitting of Spatially Varying BRDFs. *Computer Graphics Forum* 38, 4 (2019), 193–205.
G. MÃijller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. 2005. Acquisition, Synthesis, and Rendering of Bidirectional Texture Functions. *Computer Graphics Forum* 24, 1 (2005), 83–109. https://doi.org/10.1111/j.1467-8659.2005.00830.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2005.00830.x
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, Article 267 (2018), 12 pages.
Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large steps in inverse rendering of geometry. *ACM Trans. Graph.* 40, 6, Article 248 (dec 2021), 13 pages. https://doi.org/10.1145/3478513.3480501
Baptiste Nicolet, Fabrice Rousselle, Jan Novak, Alexander Keller, Wenzel Jakob, and Thomas Müller. 2023. Recursive Control Variates for Inverse Rendering. *ACM Trans. Graph.* 42, 4, Article 62 (jul 2023), 13 pages. https://doi.org/10.1145/3592139
Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo Methods for Volumetric Light Transport Simulation. 37, 2 (May 2018), 551–576. https://doi.org/10/gd2jqq
Marios Papas, Krystle de Mesa, and Henrik Wann Jensen. 2014. A Physically-Based BSDF for Modeling the Appearance of Paper. *Computer Graphics Forum* (2014).
Benjamin Roth. 2020. Broad Leaf Bidirectional Scattering Distribution Functions (BSDFs). https://doi.org/10.21227/yjek-2059
Emmanuel Turquin. 2019. Practical multiple scattering compensation for microfacet models. *URL: https://blog. selfshadow. com/publications/turquin/ms_ comp_final. pdf* 45, 3 (2019).
Delio Vicini, SÃŽbastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 108:1–108:14. https://doi.org/10.1145/3450626.3459804
Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. 195–206. https://doi.org/10/gfz4kg
Beibei Wang, Wenhua Jin, Miloš Hašan, and Ling-Qi Yan. 2022. SpongeCake: A Layered Microflake Surface Appearance Model. *ACM Trans. Graph.* 42, 1, Article 8 (2022), 16 pages.
Lifeng Wang, Wenle Wang, Julie Dorsey, Xu Yang, Baining Guo, and Heung-Yeung Shum. 2005. Real-time rendering of plant leaves. 24, 3 (jul 2005), 712âĂŞ719. https://doi.org/10.1145/1073204.1073252
Yu-Chen Wang, Chris Wyman, Lifan Wu, and Shuang Zhao. 2023. Amortizing Samples in Physics-Based Inverse Rendering Using ReSTIR. *ACM Trans. Graph.* 42, 6, Article 214 (dec 2023), 17 pages. https://doi.org/10.1145/3618331
Andrea Weidlich and Alexander Wilkie. 2007. Arbitrarily layered micro-facet surfaces. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*. 171âĂŞ178.
Mengqi Xia, Bruce Walter, Christophe Hery, and Steve Marschner. 2020. Gaussian product sampling for rendering layered materials. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 420–435.
Jiankai Xing, Xuejun Hu, Fujun Luan, Ling-Qi Yan, and Kun Xu. 2023. Extended Path Space Manifolds for Physically Based Differentiable Rendering. In *SIGGRAPH Asia 2023 Conference Papers (SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 30, 11 pages. https://doi.org/10.1145/3610548.3618195
Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. 2022. Differentiable Rendering Using RGBXY Derivatives and Optimal Transport. *ACM Trans. Graph.* 41, 6, Article 189 (nov 2022), 13 pages. https://doi.org/10.1145/3550454.3555479
K. Yan, F. Luan, M. Hašan, T. Groueix, V. Deschaintre, and S. Zhao. 2023. PSDR-Room: Single Photo to Scene using Differentiable Rendering. In *ACM SIGGRAPH Asia 2023 Conference Proceedings*.
Ling-Qi Yan, Chi-Wei Tseng, Henrik Wann Jensen, and Ravi Ramamoorthi. 2015. Physically-Accurate Fur Reflectance: Modelling, Measurement and Rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 34, 6 (2015).
Zihan Yu, Cheng Zhang, Derek Nowrouzezahrai, Zhao Dong, and Shuang Zhao. 2022. Efficient Differentiation of Pixel Reconstruction Filters for Path-Space Differentiable Rendering. *ACM Trans. Graph.* 41, 6, Article 191 (nov 2022), 16 pages. https://doi.org/10.1145/3550454.3555500
Tizian Zeltner and Wenzel Jakob. 2018. The Layer Laboratory: A Calculus for Additive and Subtractive Composition of Anisotropic Surface Reflectance. *Transactions on Graphics (Proceedings of SIGGRAPH)* 37, 4 (July 2018), 74:1–74:14. https://doi.org/10.1145/3197517.3201321
Cheng Zhang, Zhao Dong, Michael Doggett, and Shuang Zhao. 2021. Antithetic sampling for Monte Carlo differentiable rendering. *ACM Trans. Graph.* 40, 4, Article 77 (jul 2021), 12 pages. https://doi.org/10.1145/3450626.3459783
Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 143:1–143:19.
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), 227:1–227:16.
Shuang Zhao, Ravi Ramamoorthi, and Kavita Bala. 2014. High-Order Similarity Relations in Radiative Transfer. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33, 4 (July 2014), 104:1–104:12. https://doi.org/10/f6cz6p
Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016. Downsampling Scattering Parameters for Rendering Anisotropic Media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35, 6 (Nov. 2016), 166:1–166:11. https://doi.org/10/f9cpqs