

# Adjoint Nonlinear Ray Tracing

ARJUN TEH, MATTHEW O'TOOLE, and IOANNIS GKIOULEKAS, Carnegie Mellon University, USA

Reconstructing and designing media with continuously-varying refractive index fields remains a challenging problem in computer graphics. A core difficulty in trying to tackle this inverse problem is that light travels inside such media along curves, rather than straight lines. Existing techniques for this problem make strong assumptions on the shape of the ray inside the medium, and thus limit themselves to media where the ray deflection is relatively small. More recently, differentiable rendering techniques have relaxed this limitation, by making it possible to differentiably simulate curved light paths. However, the automatic differentiation algorithms underlying these techniques use large amounts of memory, restricting existing differentiable rendering techniques to relatively small media and low spatial resolutions.

We present a method for optimizing refractive index fields that both accounts for curved light paths and has a small, constant memory footprint. We use the adjoint state method to derive a set of equations for computing derivatives with respect to the refractive index field of optimization objectives that are subject to nonlinear ray tracing constraints. We additionally introduce discretization schemes to numerically evaluate these equations, without the need to store nonlinear ray trajectories in memory, significantly reducing the memory requirements of our algorithm. We use our technique to optimize high-resolution refractive index fields for a variety of applications, including creating different types of displays (multiview, lightfield, caustic), designing gradient-index optics, and reconstructing gas flows.

CCS Concepts: • Computing methodologies → Ray tracing; Computational photography.

Additional Key Words and Phrases: eikonal equation, nonlinear ray tracing, differentiable rendering, gradient-index optics

## ACM Reference Format:

Arjun Teh, Matthew O'Toole, and Ioannis Gkioulekas. 2022. Adjoint Nonlinear Ray Tracing. *ACM Trans. Graph.* 41, 4, Article 126 (July 2022), 13 pages. <https://doi.org/10.1145/3528223.3530077>

## 1 INTRODUCTION

Computer graphics research has long been fascinated by the way light refracts through transparent materials to form beautiful, but intricate, caustics (e.g., the patterns at the bottom of a pool of water). Although typically such caustics are the result of light undergoing a discrete number of refraction events, it is also possible for light to *bend continuously* through certain objects and media. For example, the shimmer observed when looking through a mass of heated air (e.g., the heat haze observed when looking down an asphalt road on a hot day) is the result of light propagating along curved paths.

Light deflects when it encounters a change in refractive index. Most simple lenses are made from material with a constant refractive index, which results in light refracting only at the interface.

---

Authors' address: Arjun Teh, ateh@andrew.cmu.edu; Matthew O'Toole, mpotoole@cmu.edu; Ioannis Gkioulekas, igkioule@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, USA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2022 Copyright held by the owner/author(s).  
0730-0301/2022/7-ART126  
<https://doi.org/10.1145/3528223.3530077>

However, the refractive index of an object or medium can also be continuous, resulting in light continuously refracting through the medium itself. Gradient-index (GRIN) lenses, such as the Luneburg lens shown in Figure 1(a) or optical fibers used for telecommunication, use spatially-varying refractive index fields to focus or steer light. Mixing two gases with different refractive index also causes light to distort when passing through, as shown in Figure 1(b).

The theory for the geometric and radiometric properties of light paths undergoing continuous refraction is well-established in computer graphics. Given a volume with a known refractive index field as input, there are many ray tracing procedures to efficiently determine the nonlinear path of light through a medium [Ament et al. 2014; Ihrke et al. 2007; Stam and Languéon 1996]. This enables the rendering of photorealistic images of these gradient-index fields.

It is also important to consider the inverse question: for a given target intensity distribution, what is the refractive index profile that produces it? For reconstruction, we could be interested in knowing the composition of a fluid mixture, and recovering the refractive index profile would give us the information to reconstruct the flow. In the case of lens design, the goal might be to design a refractive index profile that focuses or shapes light in a particular way.

One way to solve this inverse problem is through differentiable rendering, which uses reverse-mode automatic differentiation (AD) to calculate gradients with respect to some parameters of interest. Reverse-mode AD records the computation that occurs during the simulation, then calculates the derivative of the output rays with respect to the refractive index profile. This requires a large amount of memory to store the computation graph, especially when working with 3D volumes. This means that the more accurate the simulation is, the more memory automatic differentiation will require.

We present a method for differentiating the dynamics of continuous refraction *without* constructing the computation graph. We employ the adjoint state method to derive a set of ordinary differential equations for calculating the gradient. We then show how to discretize and efficiently simulate these equations. Our technique can optimize for refractive index fields based on two types of objectives: image objectives and geometric objectives. We use our technique to explore multiple types of optimization objectives for a number of applications, including: (i) optimizing the focusing properties of GRIN optical fibers and lenses, (ii) designing novel displays based on refractive index fields, and (iii) reconstructing unknown refractive index fields from a set of images. To ensure reproducibility and facilitate follow-up work, we provide our code on the project website: [http://imaging.cs.cmu.edu/adjoint\\_nonlinear\\_tracing/](http://imaging.cs.cmu.edu/adjoint_nonlinear_tracing/).

## 2 RELATED WORK

*Nonlinear ray tracing.* Our focus is on *continuously-refractive media*, where the refractive index changes continuously from one location to another. As a consequence of the eikonal equation of geometric optics, light propagating inside such media travels along curved rays, rather than (piecewise-)linear rays [Kravtsov and Orlov 1990]. Stam and Languéon [1996] use Lagrangian optics to convert

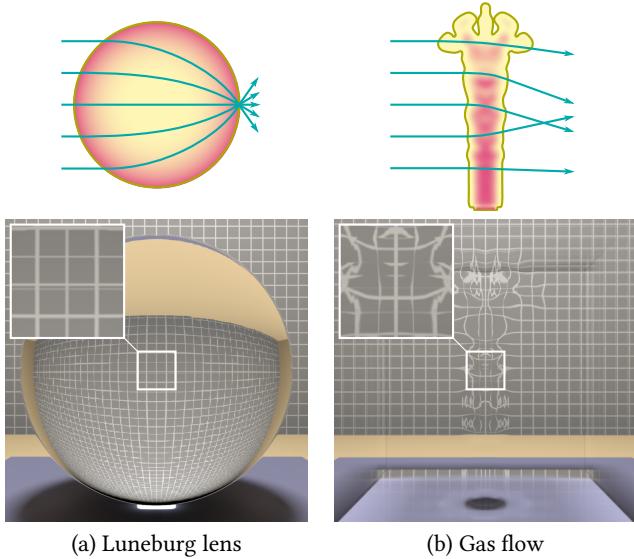


Fig. 1. Examples of refractive index fields, including a Luneburg lens and plume of gas. (a) The Luneburg lens focuses any incoming light direction to the antipodal point of the incoming direction. (b) The gas plume has a spatially-varying lower refractive index than the surrounding air, causing light to bend through the medium. To exaggerate distortion (see inset), we multiply the refractive index of the medium by 100x.

the eikonal equation into a second-order differential equation, which they then use to describe and numerically trace these curved rays. We follow Gröller [1995] and term this process *nonlinear ray tracing*. Sharma et al. [1982] reformulate these equations to make nonlinear ray tracing more numerically stable. Ihrke et al. [2007] use nonlinear ray tracing to render images due to light wavefronts propagating through continuously-refractive media. In practice, many such media additionally exhibit volumetric scattering. The combined effects of light curving and scattering can be described using the refractive radiative transfer equation [Ament et al. 2014], which can be simulated using variants of volumetric path tracing [Pediredla et al. 2020] and photon mapping [Gutierrez et al. 2003] techniques.

**Differentiable rendering.** Differentiable rendering techniques have recently emerged as an important methodology for solving inverse rendering problems; that is, problems where we are searching for scene parameters that, when used to render images, match some input measurements. A differentiable renderer can be used to calculate gradients with respect to the unknown scene parameters, which in turn can be used to perform gradient-based optimization. Recent examples of differentiable renderers include Mitsuba 2 [Nimier-David et al. 2019], Redner [Li et al. 2018], and path-space differentiable rendering (PSDR) [Zhang et al. 2020]. Importantly, Mitsuba 2 can be used to solve inverse rendering problems such as the ones we focus on, where we optimize a refractive index field subject to nonlinear ray tracing constraints. Mitsuba 2 achieves this using reverse-mode automatic differentiation. This approach allows Mitsuba 2 to generalize to a variety of inverse rendering problems. However, this approach has large memory requirements, which limits its applicability to relatively simple scenes with few unknowns. More recent

differentiable rendering techniques [Nimier-David et al. 2020; Vicini et al. 2021] overcome this problem by using a two-stage procedure that computes gradients by tracing a light path in forward and backward directions. We derive a conceptually similar procedure specifically for the case of nonlinear ray tracing.

**Design of refractive optics and caustics.** Conventional refractive optics work by refracting light a discrete number of times at interfaces with specific shapes (e.g., spherical). Recently, several works have used differentiable rendering to design the shape of these refractive interfaces, so as to optimize refractive optics for specific applications. For example, Li et al. [2021] do so for a single lens, and Sun et al. [2021] and Tseng et al. [2021] do the same to optimize a compound lens design. All these works use differentiable specular ray tracers that leverage automatic differentiation.

Other techniques for optimizing the shape of refractive interfaces have appeared for the closely-related problem of caustic design [Papas et al. 2011; Schwartzburg et al. 2014; Yue et al. 2014]. Instead of minimizing an image loss, these techniques optimize the interface shape to realize a specific mapping between incident and outgoing rays, intended to map beams incident on one side of the interface to target images on the other side.

**Gradient-index optics.** Unlike conventional refractive optics, GRIN optics work by using a continuously-varying refractive index field to guide light along curved light paths. These optics can produce optical effects and aberration characteristics that are not possible with conventional refractive optics. Teichman et al. [2013] survey GRIN optic designs, their advantages and disadvantages compared to conventional refractive optics, and ways to fabricate different GRIN profiles. As an alternative to fabrication, it is possible to sculpt *virtual* GRIN optics in various media (e.g., water, tissue) using acoustic waves [Chamanzar et al. 2019; Kang et al. 2018; Pediredla et al. 2020; Scopelliti and Chamanzar 2019; Scopelliti et al. 2020].

Two common examples of GRIN optics, for which the refractive index fields are known analytically, are the Luneburg lens [Luneberg 1944] and Maxwell fisheye lens [Maxwell 1854]. They focus collimated beams or point sources (respectively), regardless of the orientation of the lens, the angle of incidence of the beam, or the location of the point source [Kravtsov and Orlov 1990]. Another common example is GRIN waveguides, which can be used in place of conventional optical fibers [El-Diasty 2003]. Similar to our work, Balasubramanian et al. [2020] use differentiable ray tracing to design refractive index fields for new types of GRIN optics. Their technique is based on automatic differentiation, and thus suffers from slow computation and large memory consumption. Our adjoint derivation addresses both shortcomings.

**Reconstruction of transparent materials.** Some transparent materials such as gas clouds are continuously-refractive media. Atcheson et al. [2008] develop an algorithm that, given a set of image measurements of a gas cloud, solves for the refractive index everywhere inside the cloud. Their approach assumes that light rays through the cloud are approximately linear, which makes it possible to recover the refractive index field through Poisson integration. Ihrke [2007] and Ji et al. [2013] both relax the path linearity assumption by iterating between nonlinear path tracing and refractive index estimation.

However, these iterative schemes ignore either the exit direction or the exit position when path tracing. In contrast to these prior works, our method can recover the refractive index field even in cases of large deflection. Schröder and Schuster [2016] introduce theoretical analysis and an algorithm for reconstructing continuously-refractive media using time-of-flight measurements. Our method can also take advantage of time-of-flight information.

*Reverse-mode automatic differentiation.* Most existing differentiable rendering implementations use reverse-mode automatic differentiation (AD), or backpropagation [Griewank and Walther 2008]. Reverse-mode AD works as follows: First, it maintains a computation history of the program to be differentiated, in the form of a graph, until a scalar output is generated. Second, it accumulates the gradient of this output by traversing the computation graph backward using the chain rule. This two-stage procedure requires storing in memory the computation graph for the entire program. As a result, the computational complexity of the program also directly impacts the memory requirements of reverse-mode AD. In the case of nonlinear ray tracing, smaller integration step sizes result in much larger memory requirements. Our proposed method uses reversible nonlinear ray tracing that can be treated as a single node in the computation graph, resulting in constant memory use. Our approach is conceptually similar to deep learning techniques that use reversible neural network architectures to reduce the memory cost of backpropagation [Gomez et al. 2017; MacKay et al. 2018].

*Adjoint state method.* The adjoint state method is a general-purpose technique for differentiating optimization objectives that are subject to constraints that take the form of differential equations [Chavent 1974; Hinze et al. 2008]. In computer graphics, the adjoint state method has been used in problems related to rigid body dynamics and control [Geilinger et al. 2020], fluid control [McNamara et al. 2004], and surface cutting [Sharp and Crane 2018]. In differentiable rendering, Nimier-David et al. [2020] and Stam [2020] use the adjoint state method to derive algorithms that decouple the computational complexity of raytracing from the memory requirements of backpropagation. Vicini et al. [2021] use a two-stage forward and backward tracing procedure to achieve constant memory complexity during gradient calculation. We derive a conceptually similar method for light transport in continuously-refractive media.

Outside graphics, Leung et al. [2006] use the adjoint state method in seismic imaging applications, to reconstruct the refractive index of layers of the Earth based on time of flight measurements of pressure waves traveling through the ground. The propagation of such waves is governed by the same dynamic equations as the propagation of light through continuously-refractive media. However, their technique uses a formulation based on the eikonal equation and the fast marching method [Sethian 1999], whereas ours is based on Hamilton’s equations and nonlinear ray tracing. Lastly, Chen et al. [2018] use the adjoint method for differentiating ordinary differential equation systems represented as neural networks. Their method also has constant memory complexity, but uses implicit integrators for either the forward or backward pass. This necessitates using a computationally expensive nonlinear solver (e.g., Newton’s

Table 1. Definitions of main terms used in the adjoint state method.

symbol	type	description
$\sigma$	$[0, \infty)$	Parameterization of time
$\eta$	$\mathbb{R}^3 \rightarrow [1, \infty)$	Refractive index field
$\nabla\eta$	$\mathbb{R}^3 \rightarrow \mathbb{R}^3$	Spatial gradient of refractive index
Hess ( $\eta$ )	$\mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$	Hessian of the refractive index
$\mathbf{x}$	$[0, \infty) \rightarrow \mathbb{R}^3$	Position of the light ray
$\mathbf{v}$	$[0, \infty) \rightarrow \mathbb{R}^3$	Velocity of the light ray
$\lambda$	$[0, \infty) \rightarrow \mathbb{R}^3$	Adjoint state variable of position
$\mu$	$[0, \infty) \rightarrow \mathbb{R}^3$	Adjoint state variable of velocity
$C_i$	$\mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$	Inner cost function
$F_i$	$\mathbb{R} \rightarrow \mathbb{R}$	Outer cost function

method) at each integration step. We show that, by using a symplectic reversible integrator, we can make both forward and backward passes explicit, removing the need for nonlinear solvers.

### 3 THEORETICAL BACKGROUND

We use this section to present the theory of nonlinear ray tracing within media with continuously-varying refractive index, and the adjoint state method for differentiating objectives subject to partial differential equation (PDE) constraints. We refer to Kravtsov and Orlov [1990] for a more detailed discussion of nonlinear ray tracing, and to Plessix [2006] for the adjoint state method.

#### 3.1 Nonlinear ray tracing

*Nonlinear ray tracing* refers to the geometric optics description of how light propagates in *continuously-refractive media*. These are media where the refractive index  $\eta$  varies continuously from point to point. When traveling from point  $\mathbf{x}_1$  to point  $\mathbf{x}_2$  inside such a medium  $\mathcal{M} \subset \mathbb{R}^3$ , light will trace a curved ray that is a stationary point of the *optical Lagrangian*

$$L(R) \equiv \int_R \eta(\mathbf{x}(s)) ds, \quad (1)$$

where  $R$  is any curve contained in  $\mathcal{M}$  that starts at  $\mathbf{x}_1$  and ends at  $\mathbf{x}_2$ ; and  $s$  is the geometric-length (arc-length) parameterization of this ray. Equation (1) is the *optical length* of ray  $R$ , that is, geometric length weighted by the local refractive index. Therefore, light travels along curved rays that correspond to extrema (local maxima or minima) and saddle points of optical length.<sup>1</sup>

Using the Euler-Lagrange equation for stationarity of the optical Lagrangian of Equation (1), we can derive the *ray equation of geometric optics* [Born and Wolf 2013] for light rays inside a continuously-refractive medium:

$$\frac{d}{ds} \left( \eta \frac{d\mathbf{x}}{ds} \right) = \nabla\eta, \quad (2)$$

where, to simplify notation, we made the dependence of the refractive index  $\eta$  and its gradient  $\nabla\eta$  on the location  $\mathbf{x}(s) \in \mathcal{M}$  implicit.

<sup>1</sup>The optical length is directly proportional to time, given that  $dt = \eta/c_o ds$ , where  $c_o$  is the speed of light in vacuum. Therefore, stationary points of the optical Lagrangian of Equation (1) also correspond to stationary points of time.

Throughout the paper, we adopt a change of variable proposed by Sharma et al. [1982] and defined through the differential relationship

$$d\sigma \equiv \frac{ds}{\eta}. \quad (3)$$

We term  $\sigma$  the *canonical parameter*, a name we will justify shortly. By reparameterizing light rays in terms of  $\sigma$ , Equation (2) becomes

$$\frac{d}{d\sigma} \left( \frac{dx}{d\sigma} \right) = \eta \nabla \eta. \quad (4)$$

Lastly, by introducing the velocity  $v \in \mathbb{R}^3$ , we can separate this second-order ordinary differential equation (ODE) into a system of first-order ODEs known as *Hamilton's equations*,

$$\frac{dx}{d\sigma} = v, \quad (5)$$

$$\frac{dv}{d\sigma} = \eta \nabla \eta. \quad (6)$$

Equations (5)-(6) are known as the *Newton's law form* or *canonical form* of Hamilton's equations, because of their similarity with Newton's equations of motion in mechanics [Kravtsov and Orlov 1990]. We can interpret these equations as a moving particle that is subject to forces equal to  $\eta \nabla \eta$ . This exact analogy with Newton's equations of motion is a consequence of the use of  $\sigma$  from Equation 3 to parameterize light rays, justifying the name canonical parameter. As we discuss in Section 5, the use of canonical parameter  $\sigma$  also allows us to discretize and numerically simulate these equations using *reversible* symplectic integrators [Hairer et al. 2006]. The reversibility property will be critical when discretizing our adjoint nonlinear ray-tracing formulation.

### 3.2 Adjoint state method

The adjoint state method allows computing derivatives of optimization objectives subject to constraints in the form of (ordinary or partial) differential equations. Such optimization problems often arise in physics-based inverse problems. The general form of the optimization problem that we will consider is:

$$\begin{aligned} \min_{\theta} \mathcal{G}(p) \\ \text{s.t. } \mathcal{S}(p; \theta) = 0, \end{aligned} \quad (7)$$

where  $\mathcal{S}$  is the set of differential equations describing the underlying dynamics, and  $\mathcal{G}$  is the cost function of the inverse problem.  $p$  is the configuration, or *state variables*, describing the physics, and  $\theta$  is the *control variables* of the dynamics. The adjoint state method differentiates the optimization objective, making it possible to use gradient-based optimization techniques to solve optimization problems of the form shown in Equation (7).

The adjoint state method proceeds by first converting the constrained optimization problem of Equation (7) into an unconstrained optimization problem, through the use of Lagrange multipliers,

$$\min_{\theta, p, \lambda} \mathcal{L}(p, \theta, \lambda), \quad (8)$$

$$\mathcal{L}(p, \theta, \lambda) \equiv \mathcal{G}(p) - \langle \lambda, \mathcal{S}(p; \theta) \rangle. \quad (9)$$

The *Lagrangian*  $\mathcal{L}$  augments the original cost function  $\mathcal{G}$  with the original constraints, scaled by slack variables  $\lambda$  known as the *adjoint state*. The adjoint state has the same dimensionality as the configuration of the dynamics.

The Lagrangian  $\mathcal{L}$  will have the same minimum as the original cost function  $\mathcal{G}$  when all the variables except for  $\theta$  are critical points. This is equivalent to finding values  $p^*$  for the configuration and  $\lambda^*$  for the adjoint state such that

$$d_p \mathcal{L}(p^*, \theta, \lambda^*) = 0, \quad (10)$$

$$d_\lambda \mathcal{L}(p^*, \theta, \lambda^*) = 0. \quad (11)$$

Combining these equations with the definition of the Lagrangian in Equation (9), we have

$$(d_p \mathcal{S})(p^*; \theta) \lambda^* - d_p \mathcal{G}(p^*) = 0, \quad (12)$$

$$\mathcal{S}(p^*; \theta) = 0. \quad (13)$$

Equation (13) simply requires that we satisfy the constraints of the original optimization problem of Equation (7). Equation (12) allows us to solve for  $\lambda^*$ . Lastly, taking the derivative of the Lagrangian with respect to  $\theta$ , we have

$$d_\theta \mathcal{L}(p^*, \theta, \lambda^*) = -\lambda^* d_\theta \mathcal{S}(p^*; \theta). \quad (14)$$

Equation (14) equals the derivative of the constrained objective of the optimization problem of Equation (7). With this gradient in hand, we can solve this problem using any gradient-based optimization algorithm. Importantly, Equation (12) will be a set of differential equations defined by the derivatives of the original differential equations  $\mathcal{S}$  with respect to the configuration  $p$ . We can use these equations to compute  $\lambda^*$  first, and use the result to compute the product  $\lambda^* d_\theta \mathcal{S}(p^*; \theta)$  directly, without having to explicitly construct the, typically very high-dimensional, *Jacobian*  $d_\theta \mathcal{S}(p^*; \theta)$ .

### 4 DIFFERENTIATING W.R.T. REFRACTIVE INDEX

In this section, we use the adjoint state method to derive an expression for differentiating optimization objectives constrained by Hamilton's equations (5)-(6). Concretely, we are concerned with optimization problems of the form:

$$\begin{aligned} \min_{\eta} \sum_{i=1}^N \mathcal{F}_i \left[ \iint_{(\mathbf{x}_0, \mathbf{v}_0) \in \Omega} C_i \left( \mathbf{x}(\sigma_f; \eta, \mathbf{x}_0, \mathbf{v}_0), \mathbf{v}(\sigma_f; \eta, \mathbf{x}_0, \mathbf{v}_0) \right) d\mathbf{x}_0 d\mathbf{v}_0 \right] \\ \text{s.t. } \dot{\mathbf{x}}(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0) = \mathbf{v}, \quad \forall \sigma \in [0, \sigma_f], \\ \dot{\mathbf{v}}(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0) = \eta \nabla \eta, \quad \forall \sigma \in [0, \sigma_f], \\ \mathbf{x}(0; \eta, \mathbf{x}_0, \mathbf{v}_0) = \mathbf{x}_0, \\ \mathbf{v}(0; \eta, \mathbf{x}_0, \mathbf{v}_0) = \mathbf{v}_0, \end{aligned} \quad (15)$$

where the dot refers to differentiation with respect to the canonical parameter  $\sigma$ ;  $\Omega$  is the set of possible initial conditions for light rays (e.g., their location and velocity on a light source);  $\sigma_f$  parameterizes the end of the ray (e.g., when it exits the medium or reaches a sensor);  $C_i$  is any function of the location and velocity at the end of the ray; and  $\mathcal{F}_i$  is any function of integrals over multiple rays. For simplicity, we will generally use  $\mathbf{x}$  and  $\mathbf{v}$  in place of the full  $\mathbf{x}(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0)$  and  $\mathbf{v}(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0)$ . Moreover, we will use  $\mathbf{x}(\sigma_f)$  and  $\mathbf{v}(\sigma_f)$  when referring to the ray's end position and velocity, respectively.

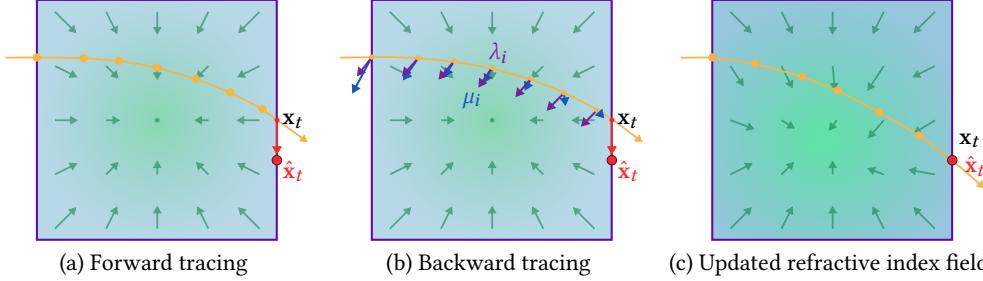


Fig. 2. The adjoint tracing procedure involves three key steps. (a) First, the ray is traced forward, where each vertex is an integration step in the simulation. An error value is computed, e.g., the distance between the terminal position  $x_t$  and a target position  $\hat{x}_t$ . (b) This error is then backward traced through the volume to compute the optimization gradient. Because of the reversibility property of the forward tracing procedure, we can retrace the same set of vertices *without* actually storing them—which would otherwise be extremely memory demanding. (c) Finally, the refractive index field  $\eta$  is updated using the gradient. This cycle is repeated until convergence. Each point on the trajectory of the ray has an associated  $\mathbf{x}$ ,  $\mathbf{v}$ ,  $\lambda$ , and  $\mu$ .

In the simplest case of the optimization problem of Equation (15), rays travel for some fixed amount of time  $\sigma_f$ , specified as part of the problem description. In practice, this time may be defined implicitly, e.g., as the time when the ray exits the medium or crosses through some specific surface inside the medium. In this case,  $\sigma_f$  is itself a function of the unknown refractive index field  $\eta$ . In both cases, differentiation reduces to solving the same pair of ordinary differential equations (Equations (19)-(20)), but with different boundary conditions. Thus, for simplicity, our derivation in the main paper assumes that  $\sigma_f$  is fixed and independent of  $\eta$ , and we show a derivation for the more general case in the supplement.

Our formulation will allow us to compute the derivative of optimization objectives such as Equation (15) with respect to refractive index, assuming derivatives of functions  $\mathcal{F}_i$  and  $C_i$  are available (e.g., through analytic or automatic differentiation). We explore two main types of losses as specializations of the optimization problem of Equation (15): image losses and geometric losses.

For image losses, the optimization problem of Equation (15) searches for a refractive index field  $\eta$  such that rendered radiometric measurements through that field match target measurements by a sensor. In this case,  $C_i$  becomes the path contribution function; the integral corresponds to a radiometric path integral expression [Veach 1998]; and  $\mathcal{F}_i$  is a loss function that compares the rendered and target radiometric measurements. An example image loss is:

$$\mathcal{F}_i = \left\| \hat{I}_i - \iint_{(\mathbf{x}_0, \mathbf{v}_0) \in \Omega} W_{e,i}(\mathbf{x}(\sigma_f), \mathbf{v}(\sigma_f)) L_e(\mathbf{x}_0, \mathbf{v}_0) d\mathbf{x}_0 d\mathbf{v}_0 \right\|^2. \quad (16)$$

In this example,  $C_i$  equals the product of the sensitivity function  $W_{e,i}$  of the  $i$ -th sensor and the source emission function  $L_e$ , and  $\mathcal{F}_i$  equals the  $\mathcal{L}_2$  loss between the rendered measurement and the actual measurement  $\hat{I}_i$  at that sensor. Summing over all  $N$  sensors (e.g., all pixels of an image) completes the loss.

For geometric losses, the optimization problem of Equation (15) searches for a refractive index field  $\eta$  such that rays traced through the field have end conditions satisfying specified properties. In this case,  $C_i$  becomes a loss function that compares the end conditions of the ray against the desired properties; the integral accumulates this loss for all rays;  $N = 1$ ; and  $\mathcal{F}$  is the identity function. An example

of a geometric loss is:

$$C_i = \left\| \mathbf{x}(\sigma_f) - \hat{\mathbf{x}} \right\|^2, \quad (17)$$

where  $C$  is the  $\mathcal{L}_2$  loss between the ray's end position  $\mathbf{x}(\sigma_f)$  and a target position  $\hat{\mathbf{x}}$  where we want all rays to arrive.

*Adjoint nonlinear ray tracing.* We now apply the adjoint state method to compute the derivative of the optimization objective of Equation (15). For simplicity, we differentiate the function inside the integral of Equation (15), which we denote as  $C$ . The total derivative additionally requires the term  $\frac{d\mathcal{F}}{dC}$ , which is easy to compute.

Using the terminology of Section 3.2, the configuration variables are  $\mathbf{p} \equiv (\mathbf{x}, \mathbf{v})$ , and the control variable is  $\theta \equiv \eta$ . Therefore, we introduce a pair of adjoint state variables  $(\lambda, \mu)$  that have the same dimensionality as the corresponding configuration variables. We can then form the Lagrangian as:

$$\mathcal{L} = C(\mathbf{x}(\sigma_f), \mathbf{v}(\sigma_f)) - \int_0^{\sigma_f} \lambda^\top (\dot{\mathbf{x}} - \mathbf{v}) d\sigma - \int_0^{\sigma_f} \mu^\top (\dot{\mathbf{v}} - \eta \nabla \eta) d\sigma. \quad (18)$$

Solving for the critical points of the Lagrangian with respect to  $\lambda$  and  $\mu$ , we obtain Hamilton's Equations (5)-(6) with initial conditions  $\mathbf{x}_0, \mathbf{v}_0$ —that is, the constraints of the optimization problem of Equation (15), as expected. Solving for the critical points with respect to  $\mathbf{x}$  and  $\mathbf{v}$ , we have

$$\dot{\lambda} = -(\nabla \eta (\nabla \eta)^\top + \eta \text{Hess}(\eta)) \mu, \quad \forall \sigma \in [0, \sigma_f] \quad (19)$$

$$\dot{\mu} = -\lambda, \quad \forall \sigma \in [0, \sigma_f] \quad (20)$$

$$\lambda(\sigma_f) = \frac{\partial C}{\partial \mathbf{x}}, \quad (21)$$

$$\mu(\sigma_f) = \frac{\partial C}{\partial \mathbf{v}}. \quad (22)$$

We provide the details of this derivation in the supplement. We make the following observations. First, Equations (19)-(20) are a system of first-order ODEs on the adjoint state variables, with boundary conditions specified by Equations (21)-(22) at the propagation *end*,  $\sigma_f$ . We term Equations (19)-(20) the *adjoint equations*. Second, computing the boundary conditions in Equations (21)-(22) requires knowing the

ray's end position  $\mathbf{x}(\sigma_f)$  and velocity  $\mathbf{v}(\sigma_f)$ . Third, even though the ODEs in Equations (19)-(20) evolve only the adjoint state variables  $\lambda, \mu$ , they require evaluating the refractive index  $\eta$  and its derivatives at all intermediate ray locations  $\mathbf{x}(\sigma), \forall \sigma \in [0, \sigma_f]$ .

These observations suggest the following two-stage procedure for computing the adjoint state variables. At the first stage, we evolve Hamilton's Equations (5)-(6) with initial conditions  $\mathbf{x}_0, \mathbf{v}_0$  forward in  $\sigma$ , until propagation ends at  $\sigma_f$ . At the second stage, we first use the ray's end position and velocity to compute the initial conditions  $\mu(\sigma_f), \lambda(\sigma_f)$  of Equations (21)-(22). We then evolve the adjoint Equations (19)-(20) with initial conditions  $\mu(\sigma_f), \lambda(\sigma_f)$  backward in  $\sigma$ , for which we travel in reverse along the ray we traced during the first stage. We refer to the second stage procedure as *backward tracing*. In Section 5, we leverage the special structure of Hamilton's equations and the adjoint equations, and devise discrete numerical procedures for efficiently implementing this two-stage procedure, without the need to store the trajectory of the ray.

Once we have computed the adjoint state variables, we can compute the gradient of the objective of Equation (15) as

$$d_\eta \mathcal{L} = \int_0^{\sigma_f} (\eta \nabla (d\eta) + d\eta \nabla \eta)^\top \boldsymbol{\mu} d\sigma. \quad (23)$$

Figure 2 visualizes the steps of the overall procedure. The differential term  $d\eta$  in Equation (23) will depend on the spatial representation of the refractive index field. We assume that we compute the refractive index using a function  $\mathcal{N}$  with parameters  $\theta, \eta(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \theta)$ . Then, we can replace  $d\eta(\mathbf{x}) = \frac{d\mathcal{N}(\mathbf{x}; \theta)}{d\theta} d\theta$ . We note that Equation (23) requires also computing spatial gradients of  $d\eta$ , i.e., the derivatives of the underlying refractive index field representation. Thus, we can use Equation (23) to compute derivatives of the objective of Equation (15) with respect to the parameters of any representation of the refractive index field that: (i) supports point, gradient (for Equations (6) and (20)), and Hessian (for Equation (20)) queries; and (ii) has point and gradient queries that are differentiable with respect to the representation parameters  $\theta$ . We refer to the supplement for the derivation of these equations for the case where  $\mathcal{N}$  corresponds to trilinear interpolation. Other representations that satisfy these requirements include smooth interpolation schemes (linear, spline, and so on), different grid types, and neural fields [Xie et al. 2021].

## 5 DISCRETIZATION OF THE ADJOINT EQUATIONS

We now discuss how to numerically implement the two-stage procedure we derived in Section 4 for differentiating the optimization objective of Equation (15) with respect to the refractive index field.

*Forward tracing.* During the first stage of our procedure, we need to evolve Hamilton's Equations (5)-(6) forward in  $\sigma$ . We choose to use a *symplectic and reversible integrator* to perform this numerical integration. When applied to a Hamiltonian system of ODEs, symplectic integrators have well-documented stability properties that help keep discretization error bounded even along very long integration trajectories. Additionally, reversible integrators will be important during the second stage of our procedure when we perform backward tracing, as we discuss later in this section. Many

---

### ALGORITHM 1: Forward Tracing

---

```

input :  $\eta, \mathbf{x}_0, \mathbf{v}_0, \Delta\sigma$ 
output:  $\mathbf{x}_f, \mathbf{v}_f$ 

 $\mathbf{x} \leftarrow \mathbf{x}_0$ 
 $\mathbf{v} \leftarrow \mathbf{v}_0$ 
while insideVolume( $\mathbf{x}$ ) do
     $\eta, \nabla\eta \leftarrow \text{interpolate}(\eta, \mathbf{x})$ 
     $\mathbf{v} \leftarrow \mathbf{v} + \eta \cdot \nabla\eta \cdot \Delta\sigma$ 
     $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v} \cdot \Delta\sigma$ 
end
 $\mathbf{x}_f \leftarrow \mathbf{x}$ 
 $\mathbf{v}_f \leftarrow \mathbf{v}$ 

```

---

### ALGORITHM 2: Backward Tracing

---

```

input :  $\eta, \mathbf{x}_f, \mathbf{v}_f, \delta\mathbf{x}, \delta\mathbf{v}, \Delta\sigma$ 
output:  $\delta\theta$ 

 $\mathbf{x} \leftarrow \mathbf{x}_f$ 
 $\mathbf{v} \leftarrow \mathbf{v}_f$ 
 $\mu \leftarrow \delta\mathbf{v}$ 
 $\lambda \leftarrow \delta\mathbf{x} + \delta\mathbf{v}\Delta\sigma$ 
 $\delta\theta \leftarrow 0$ 
while insideVolume( $\mathbf{x}$ ) do
     $\mathbf{x} \leftarrow \mathbf{x} - \mathbf{v} \cdot \Delta\sigma$ 
     $\eta, \nabla\eta \leftarrow \text{interpolate}(\eta, \mathbf{x})$ 
     $\mathbf{v} \leftarrow \mathbf{v} - \eta \cdot \nabla\eta \cdot \Delta\sigma$ 
     $nxx \leftarrow \text{Hess}(\eta, \mathbf{x})$ 
     $\lambda \leftarrow \lambda + (\nabla\eta \cdot \nabla\eta^\top + \eta \cdot nxx) \mu \Delta\sigma$ 
     $\mu \leftarrow \mu + \lambda \Delta\sigma$ 
     $\delta\theta \leftarrow \delta\theta + (\mu \cdot \nabla\eta) \frac{d\mathcal{N}}{d\theta} + \eta (\mu \cdot \nabla \frac{d\mathcal{N}}{d\theta})$ 
end

```

---

popular symplectic integrators are also reversible. For more information on symplectic and reversible integrators, we refer to Hairer et al. [2006] and Kharevych et al. [2006].

For our experiments, we use the symplectic Euler integrator, though we can easily change to other symplectic integration schemes (e.g., the leapfrog integrator [Nimier-David et al. 2019; Pediredla et al. 2020]). Applying this integrator to Hamilton's Equations (5)-(6) results in the following discretized evolution equations:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta\sigma, \quad (24)$$

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \eta(\mathbf{x}_{i-1}) \nabla\eta(\mathbf{x}_{i-1}) \Delta\sigma. \quad (25)$$

This integration scheme is *explicit*: the evolution equations compute the values of  $\mathbf{x}$  and  $\mathbf{v}$  at the  $i$ -th step using only their values and the values of the refractive index field  $\eta$  at the previous  $(i-1)$ -th step, and thus without the need for a nonlinear solver. Algorithm 1 summarizes our procedure for the forward tracing stage.

*Backward tracing.* During the second stage of our procedure, we need to evolve the adjoint Equations (19)-(20) backward in  $\sigma$ , along with retracing in reverse direction the ray we traced during the first stage. For this, we need to: first discretize the adjoint equations, as we did for Hamilton's equations in Equations (24)-(25); and second, evaluate both sets of discrete equations in the backward  $\sigma$  direction.

The adjoint equations share the same symplectic structure as Hamilton's equations. Therefore, we can discretize them using the same symplectic Euler integrator we used for Hamilton's equations:

$$\begin{aligned}\lambda_i &= \lambda_{i-1} - (\nabla \eta(\mathbf{x}_{i-1}) (\nabla \eta(\mathbf{x}_{i-1}))^\top \\ &\quad + \eta(\mathbf{x}_{i-1}) \text{Hess}(\eta(\mathbf{x}_{i-1}))) \mu_i \Delta\sigma.\end{aligned}\quad (26)$$

$$\mu_i = \mu_{i-1} - \lambda_{i-1} \Delta\sigma,\quad (27)$$

Then, we can invert these discrete relationships to evolve both Hamilton's equations and the adjoint equations in reverse, as required during the backward tracing stage:

$$\mathbf{x}_{i-1} = \mathbf{x}_i - \mathbf{v}_i \Delta\sigma,\quad (28)$$

$$\mathbf{v}_{i-1} = \mathbf{v}_i - \eta(\mathbf{x}_{i-1}) \nabla \eta(\mathbf{x}_{i-1}) \Delta\sigma,\quad (29)$$

$$\begin{aligned}\lambda_{i-1} &= \lambda_i \\ &\quad + (\nabla \eta(\mathbf{x}_{i-1}) (\nabla \eta(\mathbf{x}_{i-1}))^\top + \eta(\mathbf{x}_{i-1}) \text{Hess}(\eta(\mathbf{x}_{i-1}))) \mu_i \Delta\sigma,\end{aligned}\quad (30)$$

$$\mu_{i-1} = \mu_i + \lambda_{i-1} \Delta\sigma.\quad (31)$$

Equations (28)-(31) are simply rearranged versions of the forward discretized evolution Equations (24)-(27), respectively. The resulting numerical scheme is also *explicit*: the backward evolution equations update all quantities at the  $(i-1)$ -th step using only their values and the values of the refractive index field  $\eta$  at the  $(i)$ -th step. Algorithm 2 summarizes our backward tracing procedure.

We make two observations. First, we note that Equations (28)-(29) will query the exact same locations  $\mathbf{x}$  and refractive index values  $\eta$  as during forward tracing (up to numerical precision error). Therefore, we do *not* need to store the ray locations traced during forward tracing. We only need to use the final position and velocity  $(\mathbf{x}(\sigma_f), \mathbf{v}(\sigma_f))$  at the end of the forward tracing stage. Experimentally, we found that the relative difference between points on the (discretized) paths produced by forward and backward tracing is on the order of  $10^{-6}$ . As a result, the memory use of our combined forward tracing and backward tracing algorithms is constant, instead of scaling linearly with step-size and number of steps, as in conventional reverse-mode AD.

Second, we note that the explicit and exact reversibility properties of our two-stage procedure are due to two important choices we made in our formulation. The first choice is our use of a reversible symplectic integrator. Using a non-reversible integrator (e.g., Euler scheme) would result in the locations  $\mathbf{x}$  (and thus queried refractive index values  $\eta$ ) during backward tracing being different from those during forward tracing. In turn, this difference would result in biased gradient estimates.<sup>2</sup> The second choice is our use of the canonical parameterization  $\sigma$  and the ensuing Newton's law form of Hamilton's equations (5)-(6). Using a different parameterization of Hamilton's equations (e.g., arc-length  $s$ , as in Ihrke et al. [2007]) would result—even with a reversible integrator—in *implicit* backward equations, requiring expensive root finding procedures (e.g., Newton's method) to evolve. This is even though the corresponding continuous Hamilton's and adjoint equations are reversible. We demonstrate this in the supplement.

<sup>2</sup>By “biased”, we mean that the gradients computed by our procedure would not match those computed using automatic differentiation on the loss evaluation routine, which involves forward tracing.

*Step size.* During both the forward and backward tracing stages, it is important to select an appropriate step size  $\Delta\sigma$  for integration. Decreasing the step size increases the accuracy of the integration, at the cost of increased computation. In particular, in the case of our two-stage procedure, halving the step size would increase tracing computation by roughly 2×, given the need to trace the ray twice, forward and backward. A procedure based on reverse-mode AD would have a similar increase in computation, given the need to forward-trace a longer path and then parse a larger computational graph. However, such a procedure based on reverse-mode AD would have *additionally* increased memory requirements, given the need to store this larger computational graph. By contrast, our two-stage procedure has constant memory requirements. This highlights an important advantage of our two-stage procedure, especially when optimizing refractive index fields at higher resolutions (which requires smaller step sizes for accurate ray tracing).

## 6 RESULTS

We compare the performance of our technique with other differentiable rendering alternatives, in terms of both memory use and computational efficiency. Additionally, we show experiments using different cost functions and refractive index field representations, to show the diversity of applications of our framework. We show results for designing displays, optimizing GRIN optics, and reconstructing different types of objects. In the supplement, we show additional results that validate the accuracy of our computed gradients, and demonstrate the importance of the reversibility properties we discuss in Section 5.

*Implementation details.* We have created two implementations of our two-stage forward and backward tracing procedure: one in Pytorch [Paszke et al. 2019], and another in C++ using the Enoki library [Jakob 2019]. We use the two implementations to compare against reverse-mode AD, as implemented both by Pytorch's autograd and in Enoki. For our design and reconstruction experiments, we use the C++ implementation, combined with Pytorch for its gradient-based optimizers and visualization tools.

For all our results, we use the Adam optimizer [Kingma and Ba 2014], and initialize the refractive index field to be  $\eta(\mathbf{x}) = 1$  everywhere. To ensure that the recovered reconstruction is physically plausible, after every gradient descent iteration, we project  $\eta$  to be greater than or equal to 1 (projected gradient descent). We also use a multiresolution approach to accelerate optimization convergence: During optimization, we periodically double the resolution of the volume we use to represent the refractive index field. We select the step size used for tracing to always be smaller than the width of a voxel in the volume, meaning that we periodically decrease the step size during optimization. We also impose a constraint that the volume boundary has a refractive index of 1, by clamping the values at the boundary at each iteration. We run all of our experiments on an NVIDIA RTX 3090 GPU, with runtimes ranging between 10–40 minutes. We use the Mitsuba renderer with support for continuously-refractive media [Pediredla et al. 2020] for rendering visualizations of the results.

We note that, because of the finite step size used for tracing, a traced ray will end at some point past the volume boundary. We

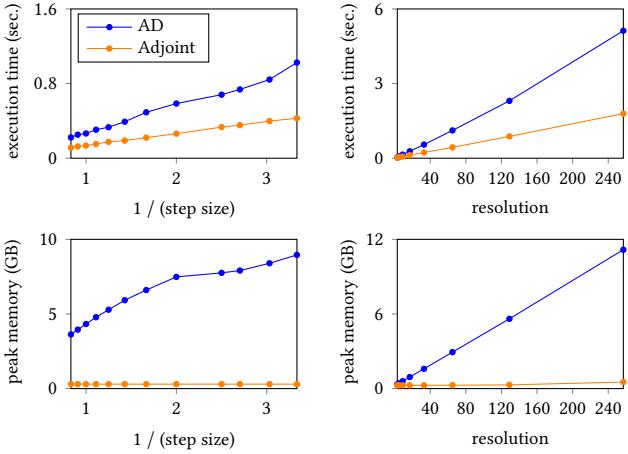


Fig. 3. Runtime and memory use comparison for reverse-mode AD and the adjoint method. The number of steps along a ray is directly proportional to resolution and inversely proportional to step size. For the adjoint method, memory use is constant and runtime increases linearly as a function of the number of steps along a ray. By contrast, memory usage and runtime for reverse-mode AD are both significantly higher. This is because reverse-mode AD requires keeping track of the entire light path to compute gradients.

deal with this by tracing the ray back to the boundary to compute the boundary conditions for the adjoint equations, then begin the backward tracing stage from the actual post-boundary end location.

**Performance.** We compare the computational efficiency and performance requirements of our technique, against the reverse-mode AD implementations of Pytorch and Enoki. In terms of memory, our method requires the initial and final positions and velocities of the ray. It also needs to keep track of a second volume that maintains the refractive index gradients  $\delta\eta$ . By contrast, reverse-mode AD additionally needs to store the computation graph, which grows with the number of integration steps taken during the simulation. This means that reverse-mode AD has a linear memory complexity with respect to step size, whereas the memory complexity of our method is constant.

To demonstrate these advantages, we perform quantitative comparisons of our C++ implementation against one that uses Enoki's reverse-mode AD. In the supplement, we show additional comparisons of our Pytorch implementation against Pytorch's reverse-mode AD. For our comparisons, we look at runtime and peak memory usage of performing forward and backward tracing operations, as a function of increasing volume resolution and decreasing step size.

Figure 3 shows the comparison results. As expected, our method has constant memory usage with respect to step size, whereas reverse-mode AD has a linear dependence on step size. Increasing the volume resolution also affects reverse-mode AD significantly more than our method. As resolution increases, the memory footprint of the volume increases by  $N^3$ . However, the step size needs to decrease along with the increase in resolution so that traced rays sample the volume voxels properly. The decrease in step size dominates the memory resources more so than the cubic increase in the

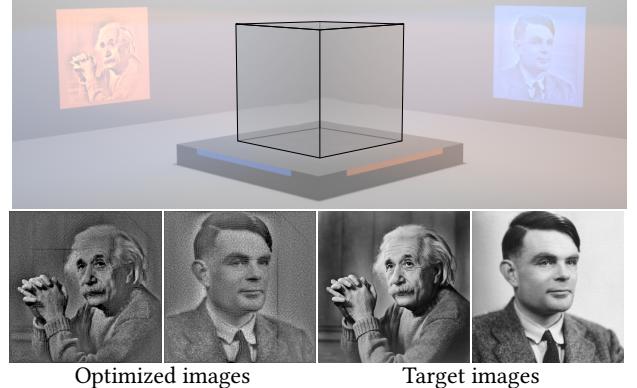


Fig. 4. The optimized GRIN lens displaying two images. **(Top)** Two collimated beams of light (red and blue) simultaneously illuminate two faces of a cubic GRIN lens, which steers the light to form two distinct images on a wall. **(Bottom)** The optimized images of Albert Einstein and Alan Turing, and the corresponding target images. The image of Albert Einstein is a portrait by Yousuf Karsh. ©Yousuf Karsh. The portrait of Alan Turing is by Elliot & Fry Studio. ©National Portrait Gallery.

volume size, which is why we see a linear dependence on resolution for reverse-mode AD as well.

Our method also performs better in terms of runtime compared to reverse-mode AD. Theoretically, our method has the same asymptotic complexity as reverse-mode AD, as traversal of the computation graph (for reverse-mode AD) and the backward tracing procedure (for our method) will both take as long as the forward tracing procedure. We attribute the better runtime observed for our method in practice to the much larger number of memory accesses that reverse-mode AD needs to perform.

We note that Enoki can perform graph simplification during automatic differentiation, which gives better memory performance than standard reverse-mode AD (so called, hybrid-mode automatic differentiation [Griewank and Walther 2008]). In our comparisons with Enoki, we keep graph simplification turned on. However, our workload requires use of the gather operation to query the spatial volume, which prevents Enoki from using certain aggressive graph simplification techniques.

**Novel view displays.** We present three experiments for designing refractive index fields that produce different displays. The first experiments is generating a multiview display that shows two images in two different directions. The second is building a multifocal display that produces images at different focusing distances with accurate defocus blur. The third is a caustic design problem where our method produces a refractive index field that generates a caustic pattern in both the near and far fields.

For the multiview display, we replicate the experimental setting presented in Nimier-David et al. [2019], where the task is to generate a refractive index volume that produces two different images simultaneously using two perpendicular light sources. The sources are collimated beams. We use an image loss in this case where, using

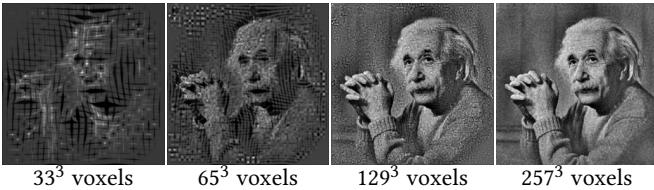


Fig. 5. Effect of volume resolution. We optimize volumes of different resolution to reproduce a picture of Albert Einstein, under the same setting as in Figure 4. As the volume resolution increases, so does reproduction accuracy. The image is courtesy of Yousuf Karsh. ©Yousuf Karsh.

the notation of Equation (15), the outer and inner cost functions are

$$\mathcal{F}_i^{\text{multiview}}(a) \equiv \|a - \hat{I}_i\|^2, \quad (32)$$

$$C_i^{\text{multiview}}(\mathbf{x}, \mathbf{v}) \equiv I_i(\mathbf{x}, \mathbf{v}), \quad (33)$$

and summation is over the two views.

Figure 4 shows the results. Our method allows us to optimize a higher-resolution volume than that used by Nimier-David et al. [2019] ( $256^3$  versus  $150^3$  voxels; trying to use  $256^3$  with the reverse-mode AD implementation in a single pass resulted in an “out-of-memory” error). We note that our experiments produce different results than those in Nimier-David et al. [2019], due to differences in the experimental setup.

Thanks to its constant memory complexity, our method enables optimizing refractive index fields of higher resolution than what is possible using reverse-mode AD. To demonstrate the importance of this capability, in Figure 5, we repeat the experiment of Figure 4 using refractive index volumes of different resolutions. Increasing the volume resolution helps ameliorate discretization artifacts due to trilinear interpolation, which in turn results in more accurate reproduction of the target image.

For the multifocal display, we optimize a refractive index field that can generate a focal stack of some input scene. We can simulate the focal stack by placing image planes at different distances from the refractive index volume, as shown in Figure 6. The distance of each plane then corresponds to projecting an image at a different focusing distance, with appropriate defocus blur. We select focusing distances equally spaced in diopter space. We then run optimization with an image loss where, using the notation of Equation (15), the outer and inner cost functions are

$$\mathcal{F}_i^{\text{multifocal}}(a) \equiv \|a - \hat{I}_i\|^2, \quad (34)$$

$$C_i^{\text{multifocal}}(\mathbf{x}, \mathbf{v}) \equiv I_i(\mathbf{x}, \mathbf{v}). \quad (35)$$

and summation is over the different focusing distances.

Figure 6 shows the results of the optimization. We form the target images by focusing the Lego knights lightfield from the Stanford light field dataset at different focusing distances. Our optimized refractive index fields can produce images that replicate the defocus blur effects in the input images.

Lastly, we design a refractive index field that generates circular caustics in both the near and far fields.<sup>3</sup> Instead of an image loss,

<sup>3</sup>By “far field” we mean the image we would obtain if we placed a sensor at a plane placed at the infinity focus of a lens. The location of rays on this plane is determined by their velocity  $\mathbf{v}$  rather than their location  $\mathbf{x}$ .

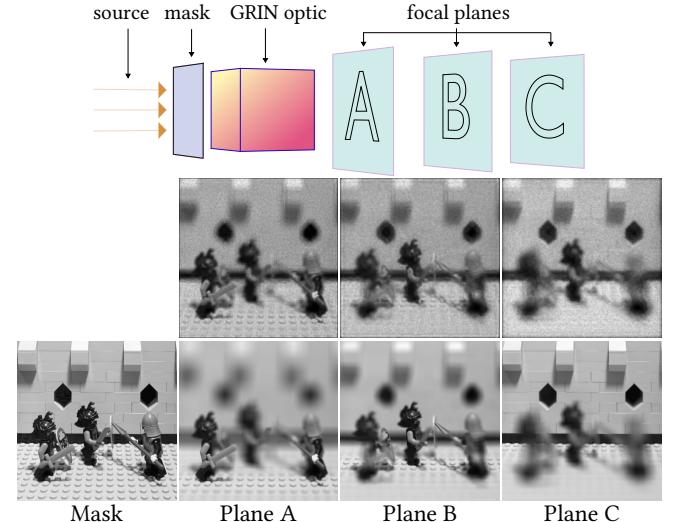


Fig. 6. A multifocal display. **(Row 1)** Collimated light passes through a mask to form an all-in-focus projected image of a lego scene. Placing a GRIN lens in front of the mask produces a 2D intensity distribution that can change as a function of distance. We optimize this GRIN lens to create a focal stack of this lego scene. **(Row 2)** The optimized intensity distribution at different plane positions, where plane A is the closest to the GRIN lens and plane C is the furthest. **(Row 3)** The target (ground truth) focal stack.

for this caustic design, we use a geometric loss that encourages rays to land at a particular locus of points and have a particular direction. The use of such a loss function showcases the ability of our method to optimize both image and geometric objectives. To define the loss, we use the signed distance function of the target caustic patterns at the near and far field planes. In the notation of Equation (15), our loss corresponds to using the outer and inner cost functions

$$\mathcal{F}^{\text{caustic}}(a) \equiv a, \quad (36)$$

$$C^{\text{caustic}}(\mathbf{x}, \mathbf{v}) \equiv (\text{SDF}_{\text{near}}(\mathbf{x}))^2 + (\text{SDF}_{\text{far}}(\mathbf{v}))^2. \quad (37)$$

To compute the loss, rays are traced until they reach the end of the volume. There, the near-field part of the loss penalizes large distances (as measured by the SDF) of the ray end-location  $\mathbf{x}$  from the target caustic. Likewise, the far-field part of the loss penalizes large distances of the ray end-velocity  $\mathbf{v}$  from the target caustic.

Figure 7 shows the results. We note that the optimized refractive index volume successfully reproduces the target caustic in both near-field and far-field. However, some parts of the caustic are a lot brighter than other parts. This is because the loss function we use encourages rays to move toward the caustic, but *not* to spread uniformly along it. Our focus in this experiment is to show that our technique can optimize geometric losses, rather than to find the best loss for producing uniformly-illuminated caustics. In fact, the general problem of defining a transport map between the source and the target images is an active area of research [Meyron et al. 2018; Schwartzburg et al. 2014; Wei et al. 2020]; our optimization procedure could be used with a geometric loss, to design refractive index field realizing such a transport map.

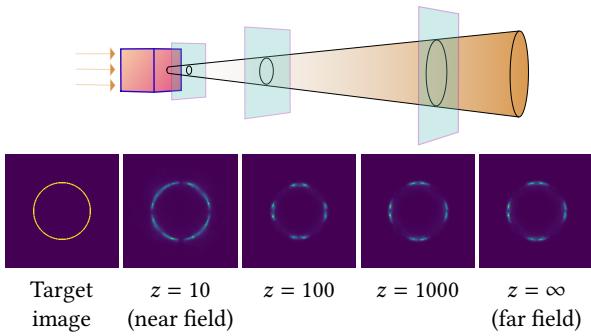


Fig. 7. Caustic design. The caustic pattern remains in shape as the sensor moves away from the volume. Since the cost function does not promote uniform energy distribution, the caustic contains bright spots.

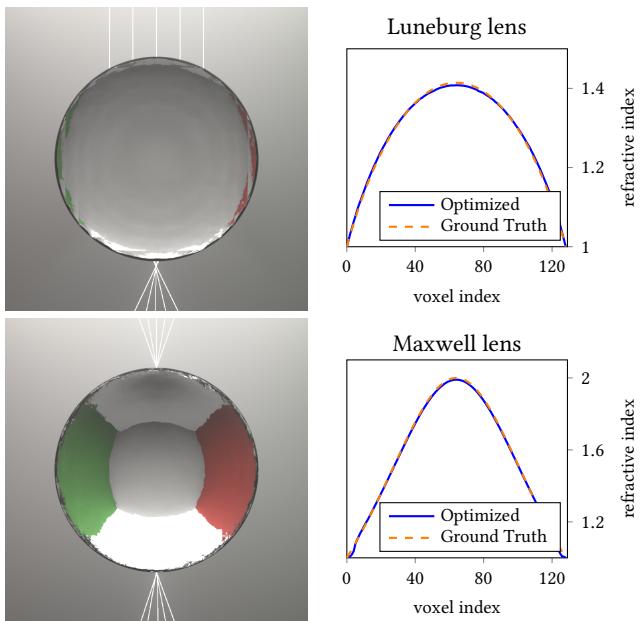


Fig. 8. Renderings of the reconstructed Luneburg and Maxwell lenses. (Col 1) Renderings of both lenses in the Cornell box with lasers shining through the lens. (Col 2) A comparison of the center axes of the optimized volume and the ground truth lens. Qualitatively, the paths have little divergence between the two. The greatest disparity occurs at the boundary of the volume. This is because of the constraint we add during optimization which projects the boundary of the solution to a value of one.

*Optimizing GRIN optics.* We show experiments where we recover the refractive index field of a known GRIN lens, using only a description of the operation of the lens (i.e., the geometric mapping between incident and outgoing rays that the lens implements). We do this for the Luneburg and Maxwell lenses. The Luneburg lens [Luneburg 1944] is a GRIN lens that focuses a point source at infinity to the antipodal point on the lens; whereas the Maxwell lens [Maxwell 1854] focuses a point source placed at the surface of the lens to the antipodal point on the lens. These perfect focusing properties are

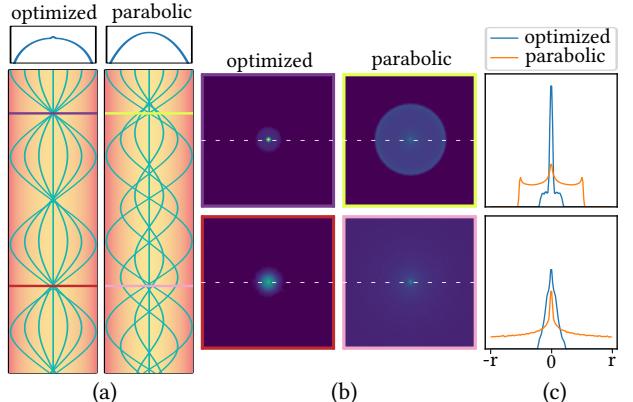


Fig. 9. A comparison between the optimized fiber design and the parabolic profile. (a) A cross-section of each fiber with the ray trajectories. Light disperses the farther it travels in both fibers, but much less so in the optimized fiber. (b) Images of the focused source at each of the focus points in the fibers. The images from the optimized fiber are better focused. (c) A cross-section of the images, showing the PSF of the fiber at each of the focus points. The optimized fiber retains better focus at the farther hop.

realized using the radial refractive index profiles

$$\eta_{\text{Luneburg}}(r) \equiv \sqrt{2 - \left(\frac{r}{R}\right)^2}, \quad (38)$$

$$\eta_{\text{Maxwell}}(r) \equiv \frac{2}{1 + \left(\frac{r}{R}\right)^2}, \quad (39)$$

where  $R$  is the radius of the lens, and  $r$  is the distance from the center of the lens. Even though the Luneburg and Maxwell lenses are primarily of theoretical interest, they provide useful groundtruth profiles for evaluating our method.

To recover these profiles, we optimize refractive index fields for the same geometric loss that, in the notation of Equation (15), uses the outer and inner cost functions<sup>4</sup>

$$\mathcal{F}^{\text{GRIN}}(a) \equiv a, \quad (40)$$

$$C_{x_0, v_0}^{\text{GRIN}}(x, v) \equiv \|x - \hat{x}(x_0, v_0)\|^2. \quad (41)$$

where  $\hat{x}(x_0, v_0)$  is the target antipodal point determined by the initial ray direction and position. In the case of the Luneburg lens, rays entering the lens with the same *direction* should reach the same end point. For the Maxwell lens, rays entering the lens from the same *point* should reach the same end point. At each gradient descent iteration, we pick at random six directions and a corresponding six target points to optimize over. Figure 8 shows the results. The optimized refractive index volumes closely match the analytic refractive fields for both the Luneburg and Maxwell lenses.

Next, we turn our attention to using our method to optimize a GRIN fiber. This is a long waveguide cable that has a rotationally-symmetric refractive index profile. As a result, light traveling through the fiber is curved toward the center of the fiber. As light travels

<sup>4</sup>We note that, for this experiment, we use a slightly more general inner cost function than in Equation (15), as we allow the inner cost function to change depending on the initial conditions of the ray. This does not affect our optimization formulation.

Table 2. Error values for the experiments in Figure 10, measured as  $\mathcal{L}_2$  relative error from the ground truth.

Method	Orig. Data	10x	100x	1000x
Adjoint [ours]	0.0014	0.014	0.29	3.8
Atcheson et al. [2008]	0.0110	0.113	1.28	19.3

through the medium and does not bounce off of the sides of the fiber, it can travel long distances with minimal loss of energy.

The drawback to this design, however, is that light traveling through the fiber experiences dispersion. Rays starting at the center of the fiber have a smaller optical distance to travel compared to rays starting farther away from the center. This results in the waveform eventually being deformed as propagation distance increases. Figure 9 visualizes this dispersion. Hisatomi et al. [2005] survey different radial profiles and their dispersion characteristics. The simplest such GRIN fiber profile is the parabolic one,

$$\eta_{\text{fiber}}(r) = \sqrt{2 - \left(\frac{r}{R}\right)^2}, \quad (42)$$

where  $r$  is the distance from the medial axis of the fiber, and  $R$  is the radius of the fiber.

We seek to design a profile that exhibits less dispersion compared to the parabolic profile. We start with a collimated source and have it focus to a point. Our cost function is

$$\mathcal{F}^{\text{fiber}}(a) \equiv a, \quad (43)$$

$$C^{\text{fiber}}(\mathbf{x}, \mathbf{v}) \equiv \|\mathbf{x} - \hat{\mathbf{p}}\|, \quad (44)$$

where  $\hat{\mathbf{p}}$  is the target focal point of the fiber. To enforce rotational symmetry, we use a refractive index field representation that specifies refractive index only as a function of radius.

Figure 9 shows the results. Using our optimized refractive index profile results in rays that focus better than using the parabolic profile, and that maintain better focus at multiple points throughout the fiber. At all focus points in the fiber, the optimized profile produces a more focused image than the parabolic profile.

*Fuel injection reconstruction.* An application for our refractive index field optimization procedure is the reconstruction of transparent gas flows, similar to Atcheson et al. [2008] and Ji et al. [2013]. Both works use active sensing to obtain measurements of the light field entering and exiting a gas volume. They generate correspondences between incident and outgoing rays; then use this information to reconstruct the refractive index field of the volume by assuming that light rays are approximately linear. This is an accurate assumption given that, as the gas flow volume has refractive index changes on the order of  $10^{-4}$ , most rays undergo very little deflection.

We reconstruct the fuel injection dataset from SFB 382 of the German Research Council (DFG) using the measurements as in these prior works. We optimize a geometric loss that, in the notation of Equation (15), uses the outer and inner cost functions

$$\mathcal{F}^{\text{fuel}}(a) \equiv a, \quad (45)$$

$$C^{\text{fuel}}_{\mathbf{x}_0, \mathbf{v}_0}(\mathbf{x}, \mathbf{v}) \equiv \|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{x}_0, \mathbf{v}_0)\|^2 + \|\mathbf{v} - \hat{\mathbf{v}}(\mathbf{x}_0, \mathbf{v}_0)\|^2, \quad (46)$$

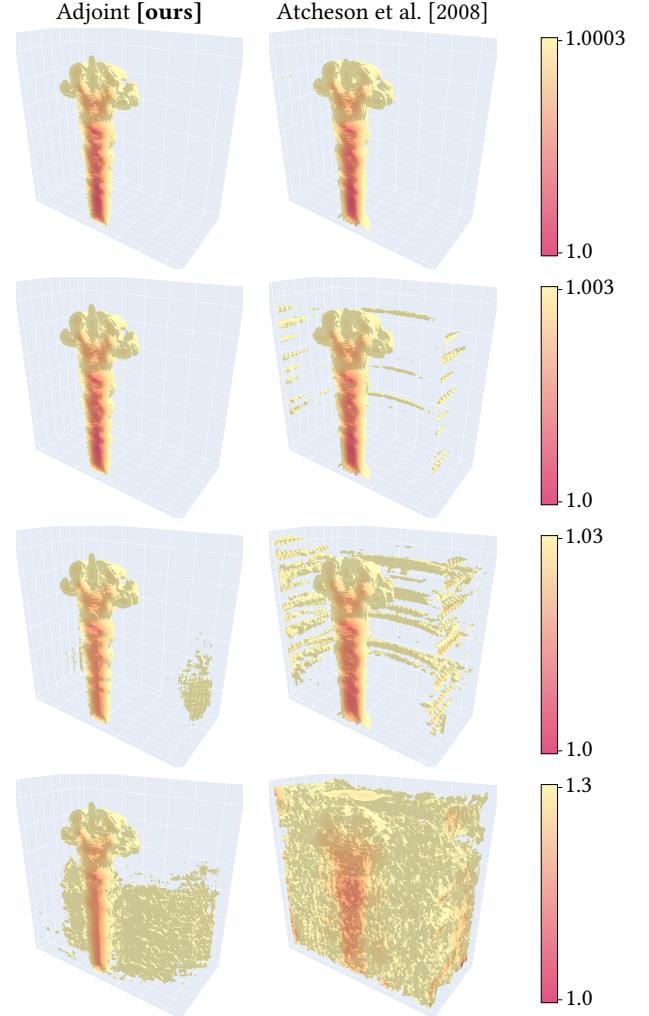


Fig. 10. Isocontour visualizations of the reconstructions of an unknown refractive index field from a set of images with our method and Atcheson et al. [2008]. Every row increases the magnitude of the refractive index field by a factor of 10. With increased refractive index gradients, the light rays deflect by larger amounts. The method proposed by Atcheson et al. [2008] struggles to recover the field in such scenarios, because of a linear path assumption imposed on the propagation of light. In contrast, our method recovers the fuel injection scene even in the case of extreme ray deflections.

where  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{v}}$  are the measurements from the ground truth simulation. Importantly, our technique does not assume that the paths of the rays are known prior to the optimization (as Ji et al. [2013] do), or linear (as Atcheson et al. [2008] do).

Figure 10 compares our results with those from the technique of Atcheson et al. [2008]. Along with experiments using the original dataset, we show experiments where we scale the refractive index values of the volume, to artificially generate scenes that produce much larger ray deflections. These artificial settings are not representative of the original application on gas flow reconstruction, but help highlight the ability of our technique to handle large deflections. As ray deflection increases, the technique of Atcheson et

al. [2008] produces reconstructions with more artifacts than in those produced by our technique. The remaining artifacts are because of the nonconvex nature of this inverse problem, and can potentially be reduced with more measurements. Table 2 compares the performance of our method and the method of Atcheson et al. [2008] in terms of relative error from the ground truth, showing that our method improves performance using the same measurements.

## 7 LIMITATIONS AND DISCUSSION

We discuss some of the limitations of our work and potential solutions that can be explored in future work.

*Initialization.* As with any gradient-based optimization procedure, our method requires a good initialization. Given that the optimization landscape is highly non-convex, it is easy for gradient descent to get trapped into local minima. In our experiments, initializing to a uniform refractive index field gave satisfactory results. Exploring better initialization schemes (e.g., reconstructions from techniques assuming a single refraction event [Atcheson et al. 2008]) is an important future research direction.

*Sufficient measurements.* When using our technique for reconstruction tasks, it is still unclear how many and what measurements are required to correctly recover the underlying refractive index field. In the presence of ambiguities, it is possible to have zero measurement loss, while still recovering a different refractive index field. The analysis of what are sufficient measurements for unique refractive index field recovery is an important open problem.

*Scattering.* Our theory and algorithms apply to media where there is only continuous refraction, and no volumetric scattering. However, most real-world materials have both continuous refraction and volumetric scattering. Currently, there exist forward rendering formulations based on the refractive radiative transfer equation for the simulation of such materials [Ament et al. 2014; Pediredla et al. 2020]. We believe our techniques can be combined with the refractive radiative transfer equation, to enable differentiable rendering in materials that both scatter and continuously refract light.

*Discretization bias.* Our procedure uses discrete numerical integration to simulate continuous ODEs. Inevitably, this introduces bias, which is larger as the simulation step size increases. However, we note that our derivation in Section 4 is continuous and thus unbiased. Investigating ways to simulate these continuous equations without discretization bias (e.g., Monte Carlo and randomization techniques) is an interesting future research direction.

*Fabrication constraints.* In most of our simulated experiments, we did not take into account possible fabrication constraints in real-world design tasks, such as those discussed by Teichman et al. [2013]. However, as we show in the GRIN fiber optimization example, it is possible to incorporate design constraints into our framework (e.g., rotational symmetry). Future work should investigate incorporating other types of design and fabrication constraints into our optimization framework.

## 8 CONCLUSION

We presented a theory for differentiating optimization objectives constrained by nonlinear ray tracing equations with respect to the underlying refractive index field. We showed both a continuous formulation, and a discretization that lends itself to numerical evaluation. The resulting algorithm has constant memory complexity, and overall requires significantly less memory than previous differentiable rendering methods based on reverse-mode automatic differentiation. Our method supports different types of optimization objectives, involving image and geometric losses. Lastly, we demonstrated the utility of our method through simulated experiments, where we use it for a variety of design and reconstruction problems involving continuously-varying refractive index fields.

## ACKNOWLEDGMENTS

We thank Adithya for his assistance with using Mitsuba for eikonal rendering. This work was supported by the National Science Foundation (NSF) under awards 1730147, 1900849, and 2008464. Ioannis Gkioulekas was supported by a Sloan Research Fellowship.

## REFERENCES

- Marcos Ament, Christoph Bergmann, and Daniel Weiskopf. 2014. Refractive radiative transfer equation. *ACM TOG* (2014).
- Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. 2008. Time-resolved 3d capture of non-stationary gas flows. *ACM TOG* (2008).
- Manushankar Balasubramanian, Sawyer D. Campbell, and Douglas H. Werner. 2020. Highly-efficient GRIN Lens Optimization Through Differential Ray Tracing. *IEEE ISAP* (2020).
- Max Born and Emil Wolf. 2013. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*.
- Maysamreza Chamanzar, Matteo Giuseppe Scopelliti, Julien Bloch, Ninh Do, Minyoung Huh, Dongjin Seo, Jillian Iafrati, Vikaas S Sohal, Mohammad-Reza Alam, and Michel M Mahabir. 2019. Ultrasonic sculpting of virtual optical waveguides in tissue. *Nature Communications* (2019).
- Guy Chavent. 1974. Identification of Functional Parameters in Partial Differential Equations. *Joint Automatic Control Conference*.
- Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural Ordinary Differential Equations. *NeurIPS* (2018).
- Fouad El-Diasty. 2003. Evaluation of some GRIN fiber parameters and the associated fraction mode loss due to mechanically induced optical anisotropy. *Applied optics* (2003).
- Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. 2020. ADD: analytically differentiable dynamics for multi-body systems with frictional contact. *ACM TOG* (2020).
- Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. 2017. The reversible residual network: Backpropagation without storing activations. *NeurIPS* (2017).
- Andreas Griewank and Andrea Walther. 2008. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM review.
- Eduard Gröller. 1995. Nonlinear ray tracing: Visualizing strange worlds. *The Visual Computer* (1995).
- Diego Gutierrez, A Muñoz, F Seron, E Jimenez, María de Luna, and Edificio Ada Byron. 2003. Global illumination in inhomogeneous media based on curved photon mapping. *Visualization, Imaging, and Image Processing* (2003).
- Ernst Hairer, Christian Lubich, and Gerhard Wanner. 2006. *Geometric numerical integration*. Springer-Verlag, Berlin.
- Michael Hinze, René Pinnau, Michael Ulbrich, and Stefan Ulbrich. 2008. *Optimization with PDE constraints*. Springer.
- Makiko Hisatomi, Michael C. Parker, and Stuart D. Walker. 2005. Comparison of zoned microstructure fiber geometries for low-dispersion waveguiding. *Journal of Lightwave Technology* (2005).
- Ivo Ihrke. 2007. Reconstruction and rendering of time-varying natural phenomena. (2007).
- Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. 2007. Eikonal rendering: Efficient light transport in refractive objects. *ACM TOG* (2007).
- Wenzel Jakob. 2019. Enoki: structured vectorization and differentiation on modern processor architectures. <https://github.com/mitsuba-renderer/enoki>.

- Yu Ji, Jinwei Ye, and Jingyi Yu. 2013. Reconstructing gas flows using light-path approximation. *IEEE CVPR* (2013).
- SeungYeon Kang, Elena Dotenko, David Amrhein, Christian Theriault, and Craig B Arnold. 2018. Ultra-high-speed variable focus optics for novel applications in advanced imaging. In *Photonic Instrumentation Engineering V*.
- Liliya Kharevych, W Wei, Yiyang Tong, Eva Kanso, Jerrold E Marsden, Peter Schröder, and Matthieu Desbrun. 2006. *Geometric, variational integrators for computer animation*. Eurographics Association.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980* (2014).
- Yu A Kravtsov and Yu I Orlov. 1990. *Geometrical optics of inhomogeneous media*. Springer.
- Shingyu Leung, Jianliang Qian, et al. 2006. An adjoint state method for three-dimensional transmission traveltime tomography using first-arrivals. *Communications in Mathematical Sciences* (2006).
- Tzu-Mao Li, Miika Aittala, Frédéric Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo ray tracing through edge sampling. *ACM TOG* (2018).
- Zongling Li, Qingyu Hou, Zhipeng Wang, Fanjiao Tan, Jin Liu, and Wei Zhang. 2021. End-to-end learned single lens design using fast differentiable ray tracing. *Optics Letters* (2021).
- Rudolf K Luneberg. 1944. *Mathematical Theory of Optics*. Providence. Brown Univ. Press.
- Matthew MacKay, Paul Vicol, Jimmy Ba, and Roger B Grosse. 2018. Reversible recurrent neural networks. *NeurIPS* (2018).
- James Clerk Maxwell. 1854. Solutions of problems. *Cambridge Dublin Math. J.* (1854).
- Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. 2004. Fluid control using the adjoint method. *ACM TOG* (2004).
- Jocelyn Meyron, Quentin Mérigot, and Boris Thibert. 2018. Light in Power: A General and Parameter-Free Algorithm for Caustic Design. *ACM TOG* (2018).
- Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *ACM TOG* (2020).
- Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM TOG* (2019).
- Marios Papas, Wojciech Jarosz, Wenzel Jakob, Szymon Rusinkiewicz, Wojciech Matusik, and Tim Weyrich. 2011. Goal-based caustics. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 503–511.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *NeurIPS* (2019).
- Adithya Pediredla, Yasin Karimi Chalmiani, Matteo Giuseppe Scopelliti, Maysamreza Chamanzar, Srinivasa Narasimhan, and Ioannis Gkioulekas. 2020. Path tracing estimators for refractive radiative transfer. *ACM TOG* (2020).
- Rene-Eduard Plessix. 2006. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International* (2006).
- Udo Schröder and Thomas Schuster. 2016. An iterative method to reconstruct the refractive index of a medium from time-of-flight measurements. *Inverse Problems* 32, 8 (jun 2016), 085009. <https://doi.org/10.1088/0266-5611/32/8/085009>
- Yuli Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. 2014. High-contrast computational caustic design. *ACM TOG* (2014).
- Matteo Giuseppe Scopelliti and Maysamreza Chamanzar. 2019. Ultrasonically sculpted virtual relay lens for in situ microimaging. *Light: Science & Applications* (2019).
- Matteo Giuseppe Scopelliti, Hengji Huang, Adithya Pediredla, Srinivasa G Narasimhan, Ioannis Gkioulekas, and Maysamreza Chamanzar. 2020. Overcoming the tradeoff between confinement and focal distance using virtual ultrasonic optical waveguides. *Optics Express* (2020).
- James A Sethian. 1999. Fast marching methods. *SIAM review* (1999).
- Anurag Sharma, Dhanwada Vizia Kumar, and Ajoy K. Ghatak. 1982. Tracing rays through graded-index media: a new method. *Applied Optics* (1982).
- Nicholas Sharp and Keenan Crane. 2018. Variational surface cutting. *ACM TOG* (2018).
- Jos Stam. 2020. Computing Light Transport Gradients using the Adjoint Method. *arXiv:2006.15059* (2020).
- Jos Stam and Eric Languérou. 1996. Ray tracing in non-constant media. In *EGSR*.
- Qilin Sun, Congli Wang, Fu Qiang, Dun Xiong, and Heidrich Wolfgang. 2021. End-to-End Complex Lens Design with Differentiable Ray Tracing. *ACM TOG* (2021).
- Jeremy Teichman, Jenny Holzer, Bohdan Balko, Brent Fisher, and Leonard Buckley. 2013. *Gradient index optics at DARPA*. Technical Report. Institute for Defense Analyses.
- Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-Francois Lalonde, and Felix Heide. 2021. Differentiable Compound Optics and Processing Pipeline Optimization for End-to-end Camera Design. *ACM TOG* (2021).
- Eric Veach. 1998. *Robust Monte Carlo methods for light transport simulation*. Stanford University.
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *ACM TOG* (2021).
- ShiLi Wei, ZhengBo Zhu, ZiChao Fan, and DingLin Ma. 2020. Least-squares ray mapping method for freeform illumination optics design. *Optica express* (2020).
- Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2021. Neural Fields in Visual Computing and Beyond. *arXiv:2111.11426* (2021).
- Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. 2014. Poisson-based continuous surface generation for goal-based caustics. *ACM Transactions on Graphics (TOG)* 33, 3 (2014), 1–7.
- Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-Space Differentiable Rendering. *ACM TOG* (2020).