Aperture-Aware Lens Design

Arjun Teh
ateh@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

Ioannis Gkioulekas
igkioule@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

Matthew O'Toole
motoole2@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

ABSTRACT

Optics designers use simulation tools to assist them in designing lenses for various applications. Commercial tools rely on finite differencing and sampling methods to perform gradient-based optimization of lens design objectives. Recently, differentiable rendering techniques have enabled more efficient gradient calculation of these objectives. However, these techniques are unable to optimize for light throughput, often an important metric for many applications.

We develop a method for calculating the gradients of optical systems with respect to both focus and light throughput. We formulate lens performance as an integral loss over a dynamic domain, which allows for the use of differentiable rendering techniques to calculate the required gradients. We also develop a ray tracer specifically designed for refractive lenses and derive formulas for calculating gradients that simultaneously optimize for focus and light throughput. Explicitly optimizing for light throughput produces lenses that outperform traditional optimized lenses that tend to prioritize for only focus. To evaluate our lens designs, we simulate various applications where our lenses: (1) improve imaging performance in low-light environments, (2) reduce motion blur for high-speed photography, and (3) minimize vignetting for large-format sensors.

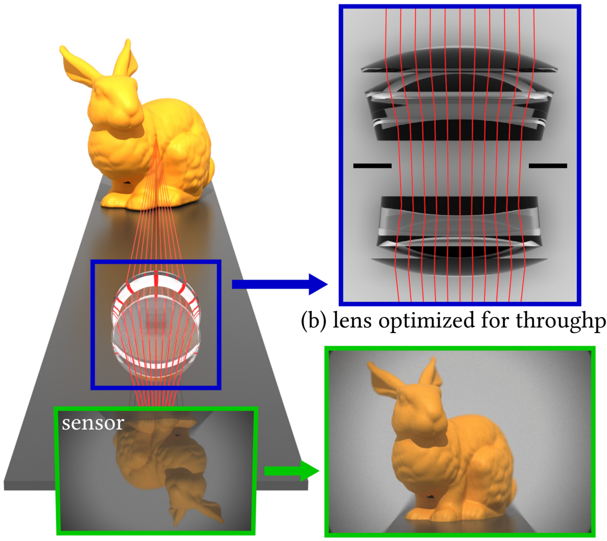
CCS CONCEPTS

• **Computing methodologies → Computational photography**; **Ray tracing**.

KEYWORDS

lens design, differentiable rendering

ACM Reference Format:
Arjun Teh, Ioannis Gkioulekas, and Matthew O'Toole. 2024. Aperture-Aware Lens Design. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641519.3657398



(a) scene layout

(b) lens optimized for throughput

(c) rendered image

Figure 1: (a) The objective of this work is to design lenses capable of forming sharp images and having high light throughput. (b) We optimize the shape and position of individual lens elements and an aperture stop to gather the light from the scene. (c) Using off-the-shelf rendering software (e.g., Blender), we then model our lens and simulate its performance in forming images under various settings.

1 INTRODUCTION

Cameras are a ubiquitous technology in our modern world. Scientists use cameras to image objects ranging from microscopic organisms to gargantuan celestial bodies. Engineers rely on cameras to collect measurements in applications ranging from remote sensing to quality assurance. Everyday consumers using cameras for entertainment or artistic pursuits. In all these settings, the imaging performance of cameras depends critically on their lenses—typically compound lenses, comprising stacks of refractive materials and opaque stops that together direct light on a sensor to form an image. In turn, the ability of a compound lens to focus light depends on the design of the geometry and materials of its optical elements.

The design space of compound lenses has been historically hard for designers to navigate, necessitating assistance from simulation and computational tools that help guide the search for performant designs. Geometric optics theory has provided a theoretical model for how light travels through the refractive materials that make up lenses [Born and Wolf 2013; Pedrotti et al. 2017]. This theory remains the basis of how optical design software such as Zemax [2023] and CODE V [2023] evaluate lens performance. Even with these tools, design remains time consuming and requires a lot of designer input. It is thus useful to develop algorithms that can further alleviate some of the burden on designers.

A common design specification is requiring a lens that takes in as much light as possible, even at the cost of reduced sharpness.

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0525-0/24/07
https://doi.org/10.1145/3641519.3657398

A light-efficient lens is especially important when imaging fast moving objects requiring a fast shutter (e.g., sports photography), or night scenes where available light is limited. However, existing design tools can only quantify the light efficiency of a lens (e.g., by computing its f-number), but cannot explicitly optimize it.

We augment the set of tools available for lens design by developing a technique to automatically optimize the light efficiency of a lens. Our key contribution is a new aperture-aware differentiable rendering technique for efficiently optimizing lens designs with respect to parameters controlling their entrance pupil. In particular, we express common lens design objectives as integrals over a domain that corresponds to the entrance pupil and depends on the lens parameters. We then derive estimators for the gradients of such objectives that correctly account for this dependence and are efficient to compute. We use our technique to explore the fundamental trade-off between light throughput and focus in lens design, and design lenses for applications where light throughput is critical, such as low-light and fast-motion settings. Our code for aperture-aware lens optimization is available at the project website.[1]

2 RELATED WORK

Lens design. Lens design has a long history in optics research and industry, and thus there are several textbooks providing detailed theory and engineering background [Pedrotti et al. 2017; Smith 2008]. Modern lens engineers use dedicated software tools for simulation and optimization, such as Zemax [2023] and CODE V [2023]. These tools enable engineers to iterate on virtual lens designs before building expensive physical prototypes.

Recent research in graphics and computational imaging has also looked at computational tools for lens design. Sun et al. [2015] search through the lens design space using a stochastic mutation policy that adds and removes surfaces from the lens. By contrast, our method focuses on improving a fixed lens design without changing its number of surfaces. Damberg and Heidrich [2015] develop a method for generating caustic images using freeform lenses. Both their method and ours rely on refractive ray tracing, but our method focuses on more general optical design problems.

Existing software tools, both industrial and research [Wang et al. 2022], support searching for lens designs that match target specifications through gradient-based optimization. Our method expands these capabilities by enabling differentiation with respect to discontinuous parameters, such as the size of physical apertures and other pupils. This allows optimizing lens designs for important metrics, such as light throughput and vignetting.

Lens modeling. Prior work in computer graphics has developed several approximate models for image formation through compound (multi-element) refractive lenses. Hullin et al. [2012] derive a model for approximating ray tracing using polynomials. Tang and Kutulakos [2013] provide a polynomial approximation of different optical aberrations in a lens. Both models trade off accuracy for efficiency, and thus are more suitable for inverse problems that require reasoning about the unknown scene that the lens is imaging. Tseng et al. [2021] have used deep learning to build differentiable approximations of the light transport inside a lens. These approximations, while efficient, do not provide sufficient accuracy for minimizing aberrations to the level required for photographic optics, and do not capture effects such as pupil size or light throughput. Our method uses geometric ray tracing, which is a more general and accurate model of light transport, and can thus optimize for such effects.

Refractive ray tracing. Our method uses the classical geometric optics model of light transport through refractive materials, which has a long history as a modeling tool for refractive lenses [Pedrotti et al. 2017]. In the context of rendering, this model translates to ray tracing with light paths that undergo a chain of specular (i.e., Dirac-delta) *refractive* events. Chen and Arvo [2000] and Walter et al. [2009] use the implicit function theorem to derive differentials for such reflective and refractive light paths with respect to their endpoints. Zeltner et al. [2020] and Jakob and Marschner [2012] use the same theory to efficiently sample and perturb specular chains for Monte Carlo rendering. Our method likewise computes differentials of specular paths, with respect to lens parameters.

Differentiable rendering. In recent years, differentiable rendering has emerged as a core methodology for solving inverse problems within vision and graphics. A differentiable renderer calculates derivatives of images, or image-based objectives, with respect to scene parameters. Examples of differentiable rendering systems include Mitsuba 3 [Jakob et al. 2022], Redner [Li et al. 2018a], and PSDR [Zhang et al. 2020]. Recent works have enabled differentiable rendering with drastically lower memory requirements [Vicini et al. 2021], efficient handling of parametric discontinuities [Bangaru et al. 2020], and support both explicit and implicit surface representations [Bangaru et al. 2022; Cai et al. 2022; Vicini et al. 2022]. Our differentiable rendering method utilizes all three of these advances, and extends support to specular-manifold light paths that comprise solely interactions with smooth refractive interfaces and optical stops—such as the paths inside a lens.

Automatic differentiation. Most differentiable rendering architectures utilize, at different stages in their pipeline, automatic differentiation (AD) (*backpropagation*) [Griewank and Walther 2008]. AD analyzes the computational graph of a program, then iteratively applies the chain rule through the graph nodes to automatically calculate gradients of the program outputs with respect to its inputs. In computer graphics, Li et al. [2018b] build a domain-specific language for differentiable programming and use it to optimize a lens design in a setting similar to ours. We also use AD in both forward and reverse mode (backpropagation), to further enable the optimization of discontinuous lens parameters such as pupil sizes.

Adjoint-state method. The adjoint-state method is a classical methodology for efficiently differentiating optimization objectives constrained by partial differential equations (e.g., the rendering equation) [Chavent 1974; Hinze et al. 2008]. More recently, Chen et al. [2018] used the adjoint-state method for differentiating ordinary differential equation systems represented as neural networks. In computer graphics, the adjoint-state method has found applications in problems including 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

[1] https://imaging.cs.cmu.edu/aperture_aware_lens_design/

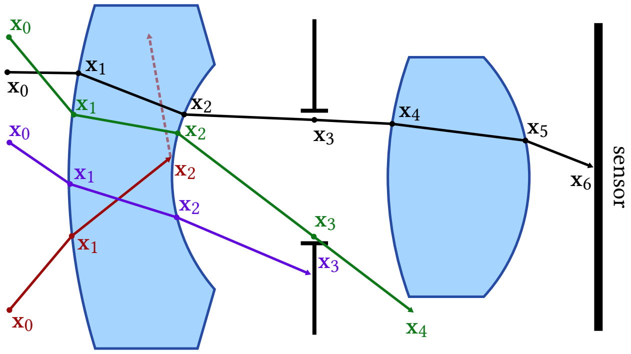


Figure 2: Rays travel through multiple refractive or aperture surfaces along the optical axis. A *valid* ray (black) travels through every surface once and in the correct order before reaching the sensor. An invalid ray either does not intersect a surface within its physical bounds (purple, green), or does so at a supercritical angle resulting in total internal reflection (red). We consider only valid rays in *sequential lens design*.

method to derive algorithms that decouple the computational complexity of ray tracing 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 derivative calculations. Similar to Wang et al. [2022] and Teh et al. [2022], we adopt an adjoint-state method for efficiently computing gradients of refractive ray tracing for lens optimization.

3 DIFFERENTIABLE RAY TRACING

Problem setting. We focus on the design of compound lenses under the following assumptions:

(1) Compound lenses comprise a given set of refractive lens singlets and mechanical aperture stops—collectively, the *lens elements*.
(2) The lens elements have a given order along an *optical axis*.
(3) All singlets are spherical lenses, and all lens elements are radially symmetric around the optical axis.

We can thus describe a compound lens as the ordered sequence of the surfaces of the lens elements—two for each singlet, one for each aperture stop. Additionally, we consider only geometric optics and focus on *sequential lens design*: that is, we consider only light rays that originate from an emitter surface before the lens, transmit once through all lens surfaces *in order*, and reach a sensor plane after the lens—we call such rays *valid rays* (Figure 2). Thus, we do not account for wave effects such as diffraction, or geometric effects involving invalid rays such as glare and interreflections.

Sequential ray tracing. Simulation of a compound lens requires tracing piecewise-linear rays through the lens' surface sequence. In turn, gradient-based optimization of a compound lens requires differentiating through this ray tracing process. To describe both primal and differentiable ray tracing, we consider a ray with initial position $\mathbf{x}_0$ on an emitter surface $\mathcal{E}$ before the first lens surface, and direction $\mathbf{v}_0$—jointly $\omega \coloneqq \{\mathbf{x}_0, \mathbf{v}_0\}$. Ray tracing sequentially applies two operations, *propagation* $\mathbf{P}$ and *transmission* $\mathbf{T}$, in alternating order at each lens surface. Given the current ray position and direction, $\mathbf{P}$ updates the ray position to the first intersection with the next lens surface; then $\mathbf{T}$ updates the ray direction at that intersection, using either Snell's law at refractive surfaces of singlets, or free-space transmission at open surfaces of aperture stops. Thus, starting at ω, ray tracing proceeds with the recurrence relation:

$$\mathbf{x}_{i+1} = \mathbf{P}\left(\mathbf{x}_i, \mathbf{v}_i, \pi\right), \tag{1}$$

$$\mathbf{v}_{i+1} = \mathbf{T}\left(\mathbf{x}_{i+1}, \mathbf{v}_i, \pi\right). \tag{2}$$

The subscript i indicates each step of the ray tracing process, as it iterates through the lens surface sequence. Ray tracing ends when the ray intersects the sensor at a position $\mathbf{x}_{N+1}$, forming a ray $\mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \ldots \mathbf{x}_N \rightarrow \mathbf{x}_{N+1}$, where N is the number of lens surfaces. Both $\mathbf{P}$ and $\mathbf{T}$ depend on the parameters π that describe the lens geometry and material properties. Thus the sequence of positions $\mathbf{x}_i$ and directions $\mathbf{v}_i$ (including $\mathbf{x}_{N+1}$) are functions of the lens parameters π and the initial conditions ω—we make this dependence explicit or implied in equations depending on context. Importantly, for a fixed lens π, the initial conditions ω completely determine, through Equations (1)–(2), a ray's position sequence $\mathbf{x}_i$. Thus we can parameterize the space of rays through the lens using the initial conditions ω, and we will refer to each ω also as a ray.

We are interested in computing the gradient of the final position $\mathbf{x}_{N+1}$ with respect to the lens parameters π. As the ray tracing process is a composition of alternating $\mathbf{P}$ and $\mathbf{T}$ operations, we can use the chain rule and compute this gradient by working backwards (i.e., *backpropagating*) through this composition. Differentiating Equations (1)–(2), we arrive at the backward recurrence:

$$\frac{\partial \mathbf{x}_{i+1}}{\partial \pi} = \frac{\partial \mathbf{P}}{\partial \mathbf{x}_i}\frac{\partial \mathbf{x}_i}{\partial \pi} + \frac{\partial \mathbf{P}}{\partial \mathbf{v}_i}\frac{\partial \mathbf{v}_i}{\partial \pi} + \frac{\partial \mathbf{P}}{\partial \pi}, \tag{3}$$

$$\frac{\partial \mathbf{v}_{i+1}}{\partial \pi} = \frac{\partial \mathbf{T}}{\partial \mathbf{x}_{i+1}}\frac{\partial \mathbf{x}_{i+1}}{\partial \pi} + \frac{\partial \mathbf{T}}{\partial \mathbf{v}_i}\frac{\partial \mathbf{v}_i}{\partial \pi} + \frac{\partial \mathbf{T}}{\partial \pi}. \tag{4}$$

At each iteration i, we can compute the gradients of $\mathbf{T}$ directly using automatic differentiation, and the gradients of $\mathbf{P}$ (an intersection operation) using implicit differentiation [Niemeyer et al. 2020].

Computing the backward recurrence in Equations (3)–(4) requires knowing the ray path. As both $\mathbf{P}$ and $\mathbf{T}$ are bijective operations, we can recover this path by starting from the final position and direction and tracing backwards. Thus, we can calculate the gradients of the final position $\mathbf{x}_{N+1}$ with respect to π without storing the ray path in memory using the following two-stage process.

Primal tracing: Starting from $\omega = \{\mathbf{x}_0, \mathbf{v}_0\}$, we use the recurrence of Equations (1)–(2) in the forward direction to compute the final ray position $\mathbf{x}_{N+1}(\omega, \pi)$ and direction $\mathbf{v}_{N+1}(\omega, \pi)$.

Adjoint tracing: Starting from $\{\mathbf{x}_{N+1}, \mathbf{v}_{N+1}\}$, we use jointly the recurrences of Equations (1)–(2) in the backward direction to retrace the ray, and Equations (3)–(4) to compute gradients.

Wang et al. [2022] derive the same process using the adjoint state method. We can analogously also compute gradients of $\mathbf{x}_{N+1}$ with respect to the initial conditions ω, which we will need in Section 5.

4 APERTURE-AWARE OPTIMIZATION

To design a compound lens, we minimize objectives that quantify its imaging performance, including objectives for blur and light efficiency. We express such objectives in the general form:

$$\mathcal{L}(\pi) \coloneqq \int_{\Omega(\pi)} \mathrm{f}(\mathbf{x}_{N+1}(\omega, \pi), \pi)\, \mathrm{d}\omega. \tag{5}$$

The scalar-valued function f describes a cost for each ray ω (e.g., squared distance of $\mathbf{x}_{N+1}$ from a target focus point). The integration is over the set of all valid rays, which we write as:[2]

$$\Omega(\pi) := \left\{\omega \in \mathcal{E}\times\mathcal{S}^2 : \mathrm{g}_i(\mathbf{x}_i(\omega,\pi), \mathbf{v}_i(\omega,\pi), \pi) < 0, i = 1, \ldots, N\right\}. \quad (6)$$

For each of the N lens surfaces, we use a scalar-valued implicit function g_i to describe the constraint that the ray must transmit through that surface—this *constraint function* is negative when that happens, and positive otherwise. As we explain next, in practice we need to specify more than one constraint function per lens surface, and Ω is the set of rays that satisfy all constraint functions for each surface. We do not explicitly indicate the multiple g_i for each i in Equation (6) to keep notation simple. We call $\Omega(\pi)$ the *entrance pupil* set, as it comprises all rays forming the entrance pupil image.

Constraint functions. To understand how to define the constraint functions, we must consider the different ways a ray may fail to transmit through a lens surface, which we show in Figure 2. As we explain later in this section, we require that these constraint functions be differentiable with respect to all three of their arguments.

We consider first a refractive surface. A ray can fail to transmit through it due to: (1) straying too far from the optical axis; or (2) experiencing total internal reflection. In the former case, the ray position on the surface exceeds the surface's physical bounds. Given radially-symmetric surfaces, we can define a constraint function:

$$\mathrm{g}_i^{\mathrm{semi}}(\mathbf{x}, \mathbf{v}, \pi) := \mathrm{dist}(\mathbf{x}) - \mathbf{R}_i(\pi), \quad (7)$$

where $\mathbf{R}_i$ is the lateral radius of the i-th surface (i.e., the size of the singlet it belongs to, determined by the lens parameters π), and $\mathrm{dist}(\mathbf{x})$ is the lateral distance of a position $\mathbf{x}$ from the optical axis.

In the latter case, the ray intersects the surface at a supercritical angle. We can define a constraint function for this case as:

$$\mathrm{g}_i^{\mathrm{TIR}}(\mathbf{x}_i, \mathbf{v}_i, \pi) := \left(\cos(\phi_i^{\mathrm{crit}}(\pi))\right)^2 - \langle \mathbf{v}_i, \hat{\mathbf{n}}(\mathbf{x}_i, \pi)\rangle^2, \quad (8)$$

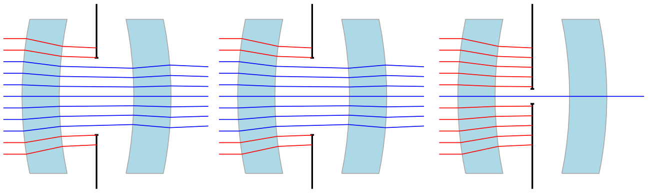
where ϕ_i^{crit} is the critical angle for the i-th surface (determined by the lens material parameters in π). We use the differences of squared cosines as, empirically, doing so improves numerical stability.

Lastly, for an aperture stop surface, we can use the same constraint functions $\mathrm{g}_i^{\mathrm{semi}}$, $\mathrm{g}_i^{\mathrm{TIR}}$, in the former with $\mathbf{R}_i$ equal to the radius of the aperture stop opening, and in the latter with $\phi_i^{\mathrm{crit}} = \pi/2$ (i.e., the constraint is always negative and thus satisfied).

Differentiating lens design objectives. Gradient-based optimization of a compound lens requires differentiating objectives as in Equation (5) with respect to the lens parameters π. If the domain of integration Ω were independent on π, this gradient would equal:[3]

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\pi} \overset{\text{biased}}{=} \int_\Omega \frac{\mathrm{d}\,\mathrm{f}}{\mathrm{d}\pi}\,\mathrm{d}\omega = \int_\Omega \frac{\partial\,\mathrm{f}}{\partial \mathbf{x}_{N+1}}\frac{\partial \mathbf{x}_{N+1}}{\partial\pi} + \frac{\partial\,\mathrm{f}}{\partial\pi}\,\mathrm{d}\omega. \quad (9)$$

We could then use automatic differentiation to compute the gradients of the cost function f, and differentiable ray tracing (Section 3) to compute the gradients of the final ray position $\mathbf{x}_{N+1}$. Techniques such as DiffOptics [Wang et al. 2022] use this approach, which we refer to as the biased gradient approach.



(a) initial design (b) biased gradients (c) unbiased gradients

Figure 3: (a) We optimize the size of an aperture stop (in black) to minimize blur. (b) Doing so using biased gradients leaves the initial lens design unaltered. (c) Using unbiased gradients results in a pinhole, and thus perfectly sharp imaging, as expected from geometric optics.

However, in general Ω *does* depend on the lens parameters π. For example, the sizes of singlets and aperture stops in a compound lens determine its entrance pupil, and thus Ω. Then, we can use the Reynolds transport theorem [Reynolds 1903] to write:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\pi} = \int_\Omega \frac{\mathrm{d}\,\mathrm{f}}{\mathrm{d}\pi}\,\mathrm{d}\omega - \int_{\partial\Omega(\pi)} \mathrm{f}\,\frac{\mathrm{d}\,\mathrm{g}^*}{\mathrm{d}\pi}\,\mathrm{d}l(\omega) \quad (10)$$

$$= \int_{\Omega(\pi)} \frac{\partial\,\mathrm{f}}{\partial \mathbf{x}_{N+1}}\frac{\partial \mathbf{x}_{N+1}}{\partial\pi} + \frac{\partial\,\mathrm{f}}{\partial\pi}\,\mathrm{d}\omega - \int_{\partial\Omega(\pi)} \mathrm{f}\left(\frac{\partial\,\mathrm{g}^*}{\partial\mathbf{x}^*}\frac{\partial\mathbf{x}^*}{\partial\pi} + \frac{\partial\,\mathrm{g}^*}{\partial\mathbf{v}^*}\frac{\partial\mathbf{v}^*}{\partial\pi} + \frac{\partial\,\mathrm{g}^*}{\partial\pi}\right)\mathrm{d}l(\omega). \quad (11)$$

In Equation (10), the *entrance pupil boundary* $\partial\Omega(\pi)$ is the space of rays that satisfy all constraint functions g_i, *except* for one *active constraint* g^* that is equal to zero. The active constraint corresponds to a lens surface where the ray barely fails to transmit—i.e., the corresponding intersection point $\mathbf{x}^*$ is exactly on the surface boundary, or direction $\mathbf{v}^*$ is incident at exactly the critical angle.

Equation (10) shows that correctly differentiating objectives for compound lens optimization requires computing the *boundary integral* over $\partial\Omega(\pi)$. We show how to efficiently do so in Section 5. But first, we use an example to demonstrate the importance of using correct gradients for compound lens optimization. We consider a compound lens comprising an aperture stop between two singlets (Figure 3(a)). We optimize only the aperture radius to minimize imaging blur, using a cost function $\mathrm{f}(\mathbf{x}_{N+1}(\omega,\pi),\pi) := \|\mathbf{x}_{N+1}(\omega,\pi) - \mathbf{o}\|^2$, where $\mathbf{o}$ is the center of the sensor plane. From geometric optics, the optimal solution involves reducing the aperture radius until we arrive at a pinhole camera, which produces a perfectly sharp image. As $\mathbf{x}_{N+1}$ does not depend on the aperture radius, the biased gradient of Equation (9) is always zero, and thus optimization using these gradients leaves the initial lens unaltered (Figure 3(b)). By contrast, optimization using the unbiased gradient of Equation (10) correctly results in a pinhole (Figure 3(c)).

5 WARP-FIELD REPARAMETERIZATION

Computing the boundary integral in Equation (10) is challenging because, for arbitrary compound lenses, the entrance pupil boundary $\partial\Omega(\pi)$ is difficult to characterize analytically or sample rays from: $\partial\Omega(\pi)$ depends on all lens elements non-linearly through ray tracing; and its shape for source points off the optical axis is nontrivial, even for radially symmetric lenses (Figure 4). We circumvent

[2] The measure $\mathrm{d}\omega$ is the product of the area $\mathrm{d}A(\mathbf{x})$ and solid angle $\mathrm{d}\sigma(\mathbf{v})$ measures.
[3] Throughout the paper, we explicitly distinguish between *total derivatives* ∂/∂ and *partial derivatives* d/d, e.g., in Equations (9)–(11) and Section 5.

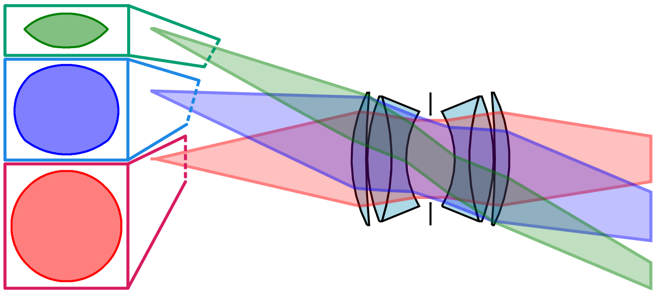


Figure 4: The shape of the entrance pupil $\Omega(\pi)$ is non-trivial even for radially symmetric lenses, becoming distorted for source points far from the optical axis. This behavior makes computing the entrance pupil boundary $\partial\Omega(\pi)$ challenging.

these challenges by using the warp-field reparameterization technique [Bangaru et al. 2022, 2020; Vicini et al. 2022] to replace the boundary integral with an integral over the entrance pupil $\Omega(\pi)$.

To this end, we first rewrite the boundary integral in Equation (10) into a form amenable to the divergence theorem as:

$$\int_{\partial\Omega(\pi)} \mathrm{f}\, \frac{\mathrm{d}\,\mathrm{g}^*}{\mathrm{d}\pi}\, \mathrm{d}l(\omega) = \int_{\partial\Omega(\pi)} \left\langle \mathrm{f}\, \frac{\mathrm{d}\,\mathrm{g}^*}{\mathrm{d}\omega}, \mathcal{V} \right\rangle \mathrm{d}l(\omega), \tag{12}$$

where the *warp field* $\mathcal{V}(\omega,\pi)$ is a vector field on Ω that must be differentiable and satisfy, as $\omega \to \partial\Omega(\pi)$,

$$\left\langle \frac{\mathrm{d}\,\mathrm{g}^*}{\mathrm{d}\omega}, \mathcal{V} \right\rangle \to \frac{\mathrm{d}\,\mathrm{g}^*}{\mathrm{d}\pi}. \tag{13}$$

Then, using the divergence theorem, Equation (12) becomes:

$$\int_{\partial\Omega(\pi)} \left\langle \mathrm{f}\, \frac{\mathrm{d}\,\mathrm{g}^*}{\mathrm{d}\omega}, \mathcal{V} \right\rangle \mathrm{d}l(\omega) = \int_{\Omega(\pi)} \mathrm{div}\,(\mathrm{f}\,\mathcal{V})\ \mathrm{d}\omega, \tag{14}$$

where div $(\cdot)$ is the divergence operator. By using Equation (14) to replace the boundary integral in Equation (10) and combining the integrands, we can write the objective gradient as:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\pi} = \int_{\Omega(\pi)} \frac{\mathrm{d}\,\mathrm{f}}{\mathrm{d}\pi} - \mathrm{div}\,(\mathrm{f}\,\mathcal{V})\ \mathrm{d}\omega. \tag{15}$$

We have thus eliminated the boundary integral, making it possible to compute unbiased Monte Carlo estimates of the objective gradient by sampling rays in the entrance pupil $\Omega(\pi)$. For each ray sample, we can evaluate the derivative terms in the integrand using differentiable ray tracing (for $\mathrm{d\,f}/\mathrm{d}\pi$, as in Section 3) and forward-mode automatic differentiation (for the warp field).

We still need to find a warp field that satisfies the stated requirements. Bangaru et al. [2022] and Vicini et al. [2022] suggest:

$$\mathcal{V}(\omega,\pi) \overset{?}{:=} \frac{\frac{\partial \mathrm{g}^*}{\partial\omega}}{\left\| \frac{\partial \mathrm{g}^*}{\partial\omega} \right\|^2} \frac{\partial\,\mathrm{g}^*}{\partial\pi}. \tag{16}$$

However, g^* is undefined inside $\Omega(\pi)$, where no constraint is active. Instead, we leverage the fact that $\mathcal{V}(\omega,\pi)$ only needs to have the correct direction as it approaches $\partial\Omega(\pi)$, and define $\mathcal{V}(\omega,\pi)$ using a mixture of such warp fields for all constraint functions:

$$\mathcal{V}(\omega,\pi) := \frac{1}{\sum_i w_i} \sum_i w_i \frac{\frac{\partial\,\mathrm{g}_i}{\partial\omega}}{\left\| \frac{\partial\,\mathrm{g}_i}{\partial\omega} \right\|^2} \frac{\partial\,\mathrm{g}_i}{\partial\pi},\ w_i(\omega,\pi) := \frac{\tanh(-\alpha\,\mathrm{g}_i)}{\mathrm{g}_i{}^p}, \tag{17}$$

with hyperparameters $p > 2$ and $\alpha > 0$. Abusing notation, the summation is over both $\mathrm{g}^{\mathrm{TIR}}$ and $\mathrm{g}^{\mathrm{semi}}$ constraint functions for each lens surface i. Each weight w_i approaches infinity as the corresponding constraint function g_i gets closer to activation; conversely, it decreases as g_i decreases. Intuitively, when a ray is close to failing to transmit through a lens surface—through either exceeding its bounds or total internal reflection—the corresponding constraint function should contribute more to the warp field.

Comparison to prior work. Vicini et al. [2022] and Bangaru et al. [2022] have used warp-field reparameterization for differentiable rendering of implicit surfaces, to deal with visibility-driven changes in the integration domain of the rendering equation. Their techniques require using a separate warp field of the form of Equation (16)—replacing our active constraint function with the implicit function defining the scene geometry—at each *stochastic* reflection or refraction event along a multi-bounce light path. By contrast, we define a single warp field for the entire multi-bounce light path. We can do so because we deal with only *specular* (i.e., Dirac-delta and therefore *deterministic*) refractions, and can thus parameterize the resulting space of specular multi-bounce paths using the initial conditions ω (Section 3). In turn, we use the same parameterization to define a multi-bounce warp field as in Equation (17). Therefore, our technique enables differentiable rendering in lower-dimensional specular path manifolds [Jakob and Marschner 2012; Zeltner et al. 2020] in the presence of parameter-dependent integration domains, which is not possible in existing differentiable rendering implementations [Jakob et al. 2022]. Even though we present our theory and technique in the context of optimizing compound refractive lenses, they should apply more broadly for other inverse problems involving specular—reflective or refractive—path manifolds, e.g., inverse rendering of specular geometry [Li et al. 2020] or caustics [Papas et al. 2011; Schwartzburg et al. 2014].

6 EXPERIMENTS

Implementation details. We implemented our warp-field technique (including the differentiable ray tracer) in JAX [Bradbury et al. 2018], and provide our implementation on the project website. We used this implementation for all our optimization experiments, for which we set $\alpha = 10$ and $p = 6$ in Equation (17), and ran Adam [Kingma and Ba 2017] for 10,000 gradient iterations. We ran all experiments on a desktop with an Intel i7-6700k CPU and NVIDIA Titan X GPU, where each experiment took approximately 10 min.

To render images with various lenses, we used either our ray tracer, or Blender [2023] with the built-in Cycles renderer. In the latter case, we modelled lens geometry as a triangle mesh.

Lens modeling and initialization. We parameterized all singlet surfaces as spherical surfaces centered on a fixed optical axis, each with four parameters: curvature, position on the optical axis, refractive index, and semidiameter. We parameterized aperture stops as planar surfaces with two parameters: position and semidiameter.

Except where we state otherwise, we initialized optimization from a double-Gauss design by Reiley [2014], which is a reconstruction of a design by Rudolph [1897]. This design has 5 singlets (10 refractive surfaces) and one aperture stop, resulting in 3×10+2 = 32 free parameters—we do not optimize refractive index.

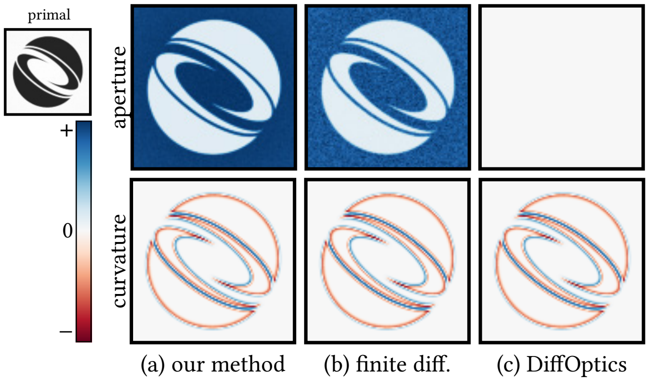


Figure 5: We consider an image of the SIGGRAPH logo through a lens, and differentiate it with respect to two lens parameters: (top row) aperture size, and (bottom row) curvature of the first refractive element. We compare the gradient images from: (a) our method, (b) finite differencing, and (c) DiffOptics [Wang et al. 2022]. Our method produces accurate gradients with respect to both lens parameters, and its results match those from finite differencing. By contrast, the biased gradients from DiffOptics cannot account for aperture size changes, producing a gradient image that is identically zero.

Finite differencing. To verify the correctness of the gradients we compute with our warp-field method, in Figure 5 we show the gradient images (also known as *forward gradients* Zhang et al. [2023]) it produces when differentiating with respect to the curvature of the first refractive surface and the semidiameter of the aperture stop of the lens. We compare these images against those from finite differencing and from the method of Wang et al. [2022], which computes biased gradients as in Equation (9). Gradients from our method match finite differencing and accurately account for changes in both curvature and aperture size; by contrast, the method of Wang et al. [2022] cannot account for aperture size changes, as those impact the integration domain $\Omega(\pi)$ rather than per-ray quantities (final ray position or throughput). Of course, even though finite differencing is useful for validating correctness, it is impractical for gradient-based optimization due to its linear complexity with parameter dimensionality, and need for step-size fine-tuning.

Light efficiency-sharpness trade-off. The example of Figure 3 highlights a classical trade-off in lens design: In the absence of diffraction effects, we can increase image sharpness by shrinking the entrance pupil of the lens (e.g., making the aperture stop smaller, or refractive surfaces flatter); however, doing so comes at the cost of decreased light efficiency, as most light rays become invalid. Conversely, increasing the entrance pupil size improves light efficiency; but it also increases geometric aberrations, resulting in decreased sharpness.

We can use our technique navigate the lens design space and achieve different trade-offs between light efficiency and sharpness. To this end, we first define two objectives of the form of Equation (5):

$$\mathcal{L}_{\text{spot}}(\pi) := \int_{\Omega(\pi)} \|\mathbf{x}_{N+1}(\omega,\pi) - \hat{\mathbf{x}}(\omega,\pi)\|^2 \,\mathrm{d}\omega, \tag{18}$$

$$\mathcal{L}_{\text{throughput}}(\pi) := \int_{\Omega(\pi)} -1 \,\mathrm{d}\omega, \tag{19}$$

where $\hat{\mathbf{x}}(\omega,\pi)$ is the centroid of the sensor locations of all rays starting from the same source point $\mathbf{x}_0 \in \omega$. The *spot-size objective* $\mathcal{L}_{\text{spot}}$ measures the spot size produced on the sensor by rays starting from the same source point. Thus, this objective, common in lens design, encourages improved sharpness. The *throughput objective* $\mathcal{L}_{\text{throughput}}$ measures the size of the entrance pupil, with a negative sign to penalize rays becoming invalid. Thus, this objective encourages improved light efficiency.

We then optimize a compound lens using the *composite objective*:

$$\mathcal{L}(\pi) := \mathcal{L}_{\text{spot}}(\pi) + \lambda\mathcal{L}_{\text{throughput}}(\pi). \tag{20}$$

The scalar weight λ allows us to control the relative importance of the spot-size and throughput objectives, and thus control how much we emphasize sharpness versus light efficiency. Gradient-based optimization of this composite objective critically depends on the ability to compute the unbiased gradient (10): the biased gradient of $\mathcal{L}_{\text{spot-size}}$ is inaccurate and, worse yet, the biased gradient (9) of $\mathcal{L}_{\text{throughput}}$ is *always zero*! This exploration of the light efficiency-sharpness tradeoff is only possible with our warp-field technique, and not those that use biased gradients [Wang et al. 2022].

Figure 6 shows an example where we optimize a 50 mm double-Gauss lens [Cox 1974] using different weights λ. Our technique produces designs that represent different trade-offs between sharpness and light efficiency. We compare our designs against those from DiffOptics [Wang et al. 2022] and Zemax [2023]: By using unbiased gradients to optimize both $\mathcal{L}_{\text{spot}}(\pi)$ and $\mathcal{L}_{\text{throughput}}(\pi)$, our warp-field technique produces lens designs that have higher throughput at comparable spot sizes, as well as designs with smaller spot sizes by giving up some throughput.

Figure 6 also shows rendered images and spot diagrams for some of the resulting lens designs.[4] We simulate the spot diagrams by focusing the lens at infinity and tracing rays parallel to the optical axis. We color the final ray points on the sensor based on how far the corresponding rays are from the optical axis. The spot diagram for the lowest λ value is much smaller than that for the highest value, indicating sharper focus; however, the resulting image is also a lot darker, indicating lower light efficiency. The figure also shows, for comparison, rendered images and spot diagrams for the designs from DiffOptics [Wang et al. 2022] and Zemax [2023].

Low-light conditions. Using a lens with high light efficiency is particularly important in low-light settings. In Figure 8, we simulate such a setting by constructing a candlelit scene. We render images of this scene using two lens designs from among those in Figure 6, one with low and another with high throughput, and incorporate Poisson and additive Gaussian noise [Hasinoff et al. 2010] in the rendered images, assuming the same exposure time. The high-throughput lens results in a visibly brighter image—a difference also visible in the image intensity histograms—with higher signal-to-noise (SNR) ratio, at the expense of some sharpness.

Motion blur. A situation where lens light efficiency is critical is in scenes with fast motion, where a high-throughput lens helps reduce exposure time, and thus motion blur. In Figure 7, we render images

[4] The rendered images are slightly mismatched, because the optimization does not explicitly enforce a focal length constraint, and thus the resulting lenses produce images with different fields of view. In our experiments, the focal length of the optimized lenses deviated by around 1 mm from that of the initial design (50 mm).

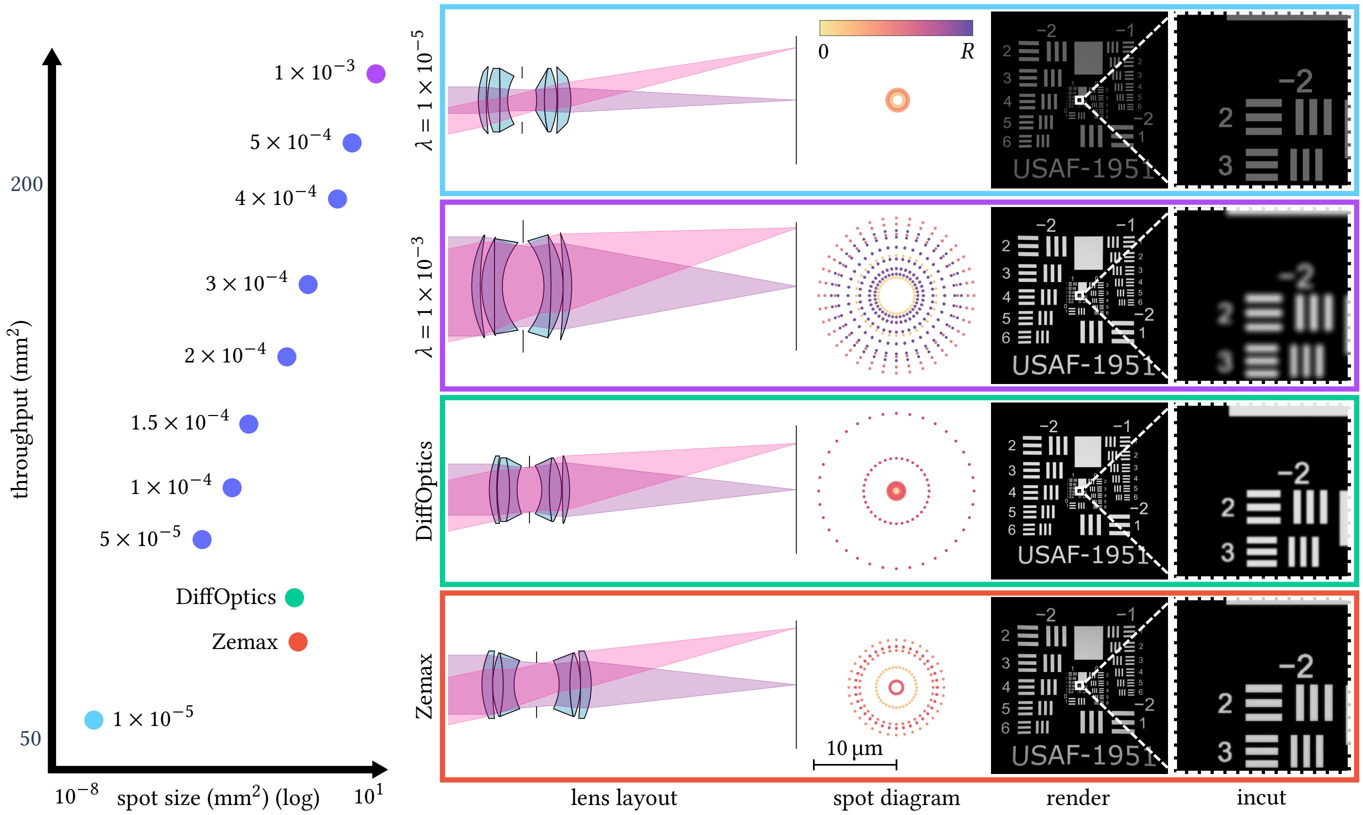


Figure 6: We use our warp-field technique to optimize the same initial design using the composite objective of Equation (20) with varying weights λ. The left plot quantifies the throughput and spot-size performance of the resulting set of lens designs, as well as designs optimized with DiffOptics [Wang et al. 2022] and Zemax [2023]. By using unbiased gradients, our technique produces lens designs with better throughput at comparable spot sizes as DiffOptics and Zemax, which both use biased gradients. Our technique additionally produces multiple other designs that achieve different trade-offs between throughput and spot size. The right figures show the lens configurations, spot diagrams, and simulated images of USAF resolution targets for the designs corresponding to the lowest (first row) and highest (second row) λ values, DiffOptics (third row) and Zemax (fourth row). These figures help qualitatively assess the differences in spot size and throughput performance. For example, the spot-size-focused design in the first row has a smaller spot, but also transmits fewer rays than the throughput-focused design in the second row.

of a scene with a USAF resolution target moving from left to right, using the same two lenses as in Figure 8. In this experiment, we adjust exposure times so that both images reach the same brightness level. The low-throughput lens requires about four times longer exposure time, resulting in significant motion blur.

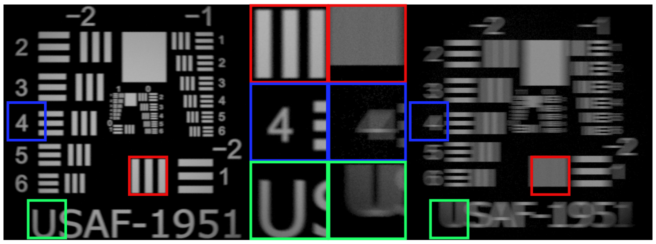
Vignetting. The composite objective of Equation (20) penalizes low throughput, and thus encourages rays starting from source points away from the optical axis to reach the sensor. As a result, using this objective, and correctly minimizing it with the unbiased gradients from our warp-field technique, helps reduce vignetting artifacts in lens designs. Figure 9 shows images of a white balance target rendered using two lenses, optimized with our warp-field technique and Zemax. As Zemax uses biased gradients, it cannot optimize for the throughput of off-axis source points, and thus its designed lens results in significant vignetting. By contrast, the lens from our technique maintains high throughput for off-axis source points, drastically reducing vignetting.

Zoom lens. In Figure 10, we use our method to improve a compound zoom lens for better sharpness and light efficiency. We start with a zoom lens design by Reiley [2014], and optimize the surface curvatures and distances, by summing the composite objectives of Equation (20) for three different focal lengths—28 mm, 45 mm, and 75 mm. Our technique produces a zoom lens design that is generally better in terms of both overall sharpness and light efficiency. At focal lengths where both designs have similar sharpness, their performance can differ significantly across the field of view; for example, at 77 mm, the optimized lens has better sharpness in the center and worse towards the edges of the image.

7 LIMITATIONS AND DISCUSSION

We discuss some limitations of our work, and potential solutions that suggest future research directions.

Nonconvexity. Our technique facilitates gradient-based optimization of complex objectives for compound lenses. However, these



high throughput weight low throughput weight

Figure 7: We simulate images of a moving USAF resolution target using lenses optimized for high throughput ($\lambda = 5 \times 10^{-4}$, left) versus small spot size ($\lambda = 1 \times 10^{-5}$, right). The high-throughput lens is well suited for this dynamic scene: it achieves the same overall brightness at one-fourth the exposure time, resulting in much reduced motion blur.

objectives are highly non-convex. Thus gradient-based optimization can get stuck in local minima. We can mitigate this issue by combining our technique with simulated annealing, or reasonable initializations from data-driven techniques [Côté et al. 2021].

Variable element number. To produce performant compound lens designs, designers typically add and remove lens elements to examine whether different configurations can improve performance. From an optimization perspective, doing so corresponds to a discrete, non-differentiable operation that changes the dimensionality of the design space. Gradient-based optimization, and thus our technique, does not accommodate such changes, but we can combine it with lens mutation techniques that do [Sun et al. 2015].

Other design constraints and trade-offs. Lens designs often need to accommodate application-specific constraints that go beyond sharpness and light efficiency. For example, the manufacturing process may constrain the realizable surface shapes. Alternatively, the intended use may constrain the lens form factor or cost. Such constraints are typically differentiable, and thus amenable to optimization using our technique. Even if they do not take the form of hard constraints, such application-specific considerations can introduce additional trade-offs that the lens design process must balance. In such cases, we can combine our technique with gradient-based multi-objective optimization techniques [Schulz et al. 2018] to discover Pareto-optimal lens designs.

Other geometric optical elements. We have not considered many geometric optical elements that are commonplace in consumer photography and scientific imaging. Aspherical, Fresnel, and freeform lenses have many more degrees of freedom than spherical ones, and are common in projector and illumination systems. Beyond dioptric (i.e., refractive) systems, catoptric (i.e., reflective) systems are common in telescopes [Gardner et al. 2006; Wilson 2013]; and catadioptric (i.e., both reflective and refractive) systems [Baker and Nayar 1999] facilitate wide-angle imaging. As such systems can be modeled using geometric optics and ray tracing, our theory directly supports or can be readily extended to support their design.

Wave optical elements. Other types of optical elements fundamentally rely on wave optics, such as diffractive optical elements and “metalenses” [Chakravarthula et al. 2023]. Such elements are becoming increasingly popular in both scientific and consumer applications, thanks to their miniaturization potential. As our theory relies on geometric optics, it does not apply to such optical elements. Recent progress in wave optics rendering [Steinberg and Yan 2021] can facilitate extensions in this direction.

Interreflections. An important effect that we did not explore is interreflections between and inside optical elements in a compound lens. In very bright scenes, such interreflections can become visible in the form of glare and lens flare. Our theory can, in principle, model this effect; however optimization of glare characteristics would require expensive Monte Carlo rendering, to account for all possible interreflections inside a lens in an unbiased manner.

Beyond cameras. We presented our technique in the context of photographic lenses, but it can apply to other domains. An example is augmented reality (AR) and virtual reality (VR), where design of eyepieces is important. VR displays use intricate configurations of optical elements [Qin et al. 2023] and benefit from high throughput designs [Damberg et al. 2016]; our technique can help optimize existing such configurations or discover new ones. Another example is non-imaging optics, i.e., optical systems designed specifically to maximize light throughput [Steinberg et al. 2022]. Our technique can be suitable for optimizing designs in this domain.

ACKNOWLEDGMENTS

We thank Yexin Hu for providing code for importing Zemax files to Blender, and Maysam Chamanzar for facilitating access to a Zemax license. This work was supported by the National Science Foundation under awards 1730147, 1900849, and 2238485, a Sloan Research Fellowship, and a gift from Google Research.

REFERENCES

Simon Baker and Shree K Nayar. 1999. A theory of single-viewpoint catadioptric image formation. *International journal of computer vision* 35 (1999), 175–196.

Sai Praveen Bangaru, Michael Gharbi, Fujun Luan, Tzu-Mao Li, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Fredo Durand. 2022. Differentiable Rendering of Neural SDFs through Reparameterization. In *SIGGRAPH Asia 2022 Conference Papers (SA ’22)*. Article 5, 9 pages. https://doi.org/10.1145/3550469.3555397

Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.* 39, 6, Article 245 (nov 2020), 18 pages. https://doi.org/10.1145/3414685.3417833

Max Born and Emil Wolf. 2013. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light.* Elsevier.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: composable transformations of Python+NumPy programs.* http://github.com/google/jax

Guangyan Cai, Kai Yan, Zhao Dong, Ioannis Gkioulekas, and Shuang Zhao. 2022. Physics-Based Inverse Rendering using Combined Implicit and Explicit Geometries. *Computer Graphics Forum* 41, 4 (2022).

Praneeth Chakravarthula, Jipeng Sun, Xiao Li, Chenyang Lei, Gene Chou, Mario Bijelic, Johannes Froesch, Arka Majumdar, and Felix Heide. 2023. Thin On-Sensor Nanophotonic Array Cameras. *ACM Trans. Graph.* 42, 6, Article 249 (dec 2023), 18 pages. https://doi.org/10.1145/3618398

Guy Chavent. 1974. Identification of Functional Parameters in Partial Differential Equations. *Joint Automatic Control Conference.*

Min Chen and James Arvo. 2000. Theory and application of specular path perturbation. *ACM Trans. Graph.* 19, 4 (oct 2000), 246–278. https://doi.org/10.1145/380666.380670

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural Ordinary Differential Equations. *NeurIPS* (2018).

Roger N. Clark. 2023. ClarkVision.com. https://clarkvision.com/articles/digital.sensor.performance.summary/index.html Accessed: 2024-13-1.

Blender Online Community. 2023. *Blender - a 3D modelling and rendering package.* Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

Geoffroi Côté, Jean-François Lalonde, and Simon Thibault. 2021. Deep learning-enabled framework for automatic lens design starting point generation. *Opt. Express* 29, 3 (Feb 2021), 3841–3854. https://doi.org/10.1364/OE.401590

Arthur Cox. 1974. *Photographic optics: a modern approach to the technique of definition.* Focal Press.

Gerwin Damberg, James Gregson, and Wolfgang Heidrich. 2016. High Brightness HDR Projection Using Dynamic Freeform Lensing. *ACM Trans. Graph.* 35, 3, Article 24 (may 2016), 11 pages. https://doi.org/10.1145/2857051

Gerwin Damberg and Wolfgang Heidrich. 2015. Efficient freeform lens optimization for computational caustic displays. *Opt. Express* 23, 8 (Apr 2015), 10224–10232. https://doi.org/10.1364/OE.23.010224

Jonathan P Gardner, John C Mather, Mark Clampin, Rene Doyon, Matthew A Greenhouse, Heidi B Hammel, John B Hutchings, Peter Jakobsen, Simon J Lilly, Knox S Long, et al. 2006. The james webb space telescope. *Space Science Reviews* 123 (2006), 485–606.

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 Trans. Graph.* 39, 6, Article 190 (nov 2020), 15 pages. https://doi.org/10.1145/3414685.3417766

Andreas Griewank and Andrea Walther. 2008. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* SIAM review.

Samuel W Hasinoff, Frédo Durand, and William T Freeman. 2010. Noise-optimal capture for high dynamic range photography. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE, 553–560.

Michael Hinze, René Pinnau, Michael Ulbrich, and Stefan Ulbrich. 2008. *Optimization with PDE constraints.* Springer.

Matthias B. Hullin, Johannes Hanika, and Wolfgang Heidrich. 2012. Polynomial Optics: A Construction Kit for Efficient Ray-Tracing of Lens Systems. *Computer Graphics Forum (Proceedings of EGSR 2012)* 31, 4 (July 2012).

Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: a Markov Chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.* 31, 4, Article 58 (jul 2012), 13 pages. https://doi.org/10.1145/2185520.2185554

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. DR.JIT: a just-in-time compiler for differentiable rendering. *ACM Trans. Graph.* 41, 4, Article 124 (jul 2022), 19 pages. https://doi.org/10.1145/3528223.3530099

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018a. Differentiable Monte Carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6, Article 222 (dec 2018), 11 pages. https://doi.org/10.1145/3272127.3275109

Tzu-Mao Li, Michaël Gharbi, Andrew Adams, Frédo Durand, and Jonathan Ragan-Kelley. 2018b. Differentiable programming for image processing and deep learning in halide. *ACM Trans. Graph.* 37, 4, Article 139 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201383

Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. 2020. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 1262–1271.

Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. 2004. Fluid control using the adjoint method. *ACM Trans. Graph.* 23, 3 (aug 2004), 449–456. https://doi.org/10.1145/1015706.1015744

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 3504–3515.

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.* 39, 4, Article 146 (aug 2020), 15 pages. https://doi.org/10.1145/3386569.3392406

Marios Papas, Wojciech Jarosz, Wenzel Jakob, Szymon Rusinkiewicz, Wojciech Matusik, and Tim Weyrich. 2011. Goal-Based Caustics. *Computer Graphics Forum (Proceedings of Eurographics)* 30, 2 (June 2011), 503–511. https://doi.org/10.1111/j.1467-8659.2011.01876.x

Frank L Pedrotti, Leno M Pedrotti, and Leno S Pedrotti. 2017. *Introduction to optics.* Cambridge University Press.

Yingsi Qin, Wei-Yu Chen, Matthew O'Toole, and Aswin C. Sankaranarayanan. 2023. Split-Lohmann Multifocal Displays. *ACM Trans. Graph.* 42, 4, Article 57 (jul 2023), 18 pages. https://doi.org/10.1145/3592110

Daniel Reiley. 2014. Lens Designs. https://www.lens-designs.com/ Accessed: 2024-04-18.

Osborne Reynolds. 1903. *Papers on Mechanical and Physical Subjects: The sub-mechanics of the universe.* Vol. 3. The University Press.

Paul Rudolph. U.S. Patent 583 336, May. 1897. Object Glass.

Adriana Schulz, Harrison Wang, Eitan Grinspun, Justin Solomon, and Wojciech Matusik. 2018. Interactive exploration of design trade-offs. *ACM Trans. Graph.* 37, 4, Article 131 (jul 2018), 14 pages. https://doi.org/10.1145/3197517.3201385

Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. 2014. High-contrast computational caustic design. *ACM Trans. Graph.* 33, 4, Article 74 (jul 2014), 11 pages. https://doi.org/10.1145/2601097.2601200

Nicholas Sharp and Keenan Crane. 2018. Variational surface cutting. *ACM Trans. Graph.* 37, 4, Article 156 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201356

Warren J Smith. 2008. *Modern optical engineering: the design of optical systems.* McGraw-Hill Education.

Jos Stam. 2020. Computing Light Transport Gradients using the Adjoint Method. *arXiv:2006.15059* (2020).

Shlomi Steinberg, Nandor Bokor, and Nir Davidson. 2022. Two-mirror compact system for ideal concentration of diffuse light. *Journal of the Optical Society of America A* 39, 4 (Mar 2022), 628. https://doi.org/10.1364/josaa.447493

Shlomi Steinberg and Ling-Qi Yan. 2021. A generic framework for physical light transport. *ACM Trans. Graph.* 40, 4, Article 139 (jul 2021), 20 pages. https://doi.org/10.1145/3450626.3459791

Libin Sun, Brian Guenter, Neel Joshi, Patrick Therien, and James Hays. 2015. Lens Factory: Automatic Lens Generation Using Off-the-shelf Components. arXiv:1506.08956 [cs.GR]

Synopsis. 2023. Code V Optical Design Software. https://www.synopsys.com/optical-solutions/codev.html

Huixuan Tang and Kiriakos N. Kutulakos. 2013. What does an aberrated photo tell us about the lens and the scene?. In *IEEE International Conference on Computational Photography (ICCP).* 1–10. https://doi.org/10.1109/ICCPhot.2013.6528316

Arjun Teh, Matthew O'Toole, and Ioannis Gkioulekas. 2022. Adjoint nonlinear ray tracing. *ACM Trans. Graph.* 41, 4, Article 126 (jul 2022), 13 pages. https://doi.org/10.1145/3528223.3530077

Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-François Lalonde, and Felix Heide. 2021. Differentiable Compound Optics and Processing Pipeline Optimization for End-to-end Camera Design. *ACM Trans. Graph.* 40, 2, Article 18 (jun 2021), 19 pages. https://doi.org/10.1145/3446791

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path replay backpropagation: differentiating light paths using constant memory and linear time. *ACM Trans. Graph.* 40, 4, Article 108 (jul 2021), 14 pages. https://doi.org/10.1145/3450626.3459804

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2022. Differentiable signed distance function rendering. *ACM Trans. Graph.* 41, 4, Article 125 (jul 2022), 18 pages. https://doi.org/10.1145/3528223.3530139

Bruce Walter, Shuang Zhao, Nicolas Holzschuch, and Kavita Bala. 2009. Single scattering in refractive media with triangle mesh boundaries. In *ACM SIGGRAPH 2009 Papers* (New Orleans, Louisiana) *(SIGGRAPH '09).* Article 92, 8 pages. https://doi.org/10.1145/1576246.1531398

Congli Wang, Ni Chen, and Wolfgang Heidrich. 2022. dO: A differentiable engine for Deep Lens design of computational imaging systems. *IEEE Transactions on Computational Imaging* 8 (2022), 905–916.

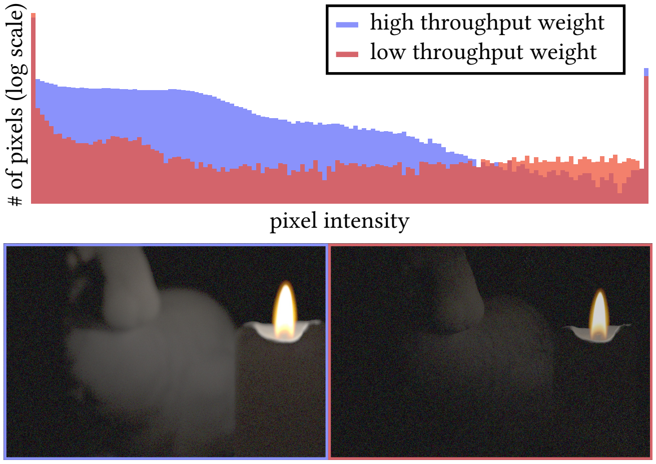
Raymond N Wilson. 2013. *Reflecting telescope optics II: manufacture, testing, alignment, Modern Techniques.* Springer Science & Business Media.

Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. 2020. Specular manifold sampling for rendering high-frequency caustics and glints. *ACM Trans. Graph.* 39, 4, Article 149 (aug 2020), 15 pages. https://doi.org/10.1145/3386569.3392408

Zemax. 2023. OpticStudio. https://www.ansys.com/products/optics/ansys-zemax-opticstudio

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4, Article 143 (aug 2020), 19 pages. https://doi.org/10.1145/3386569.3392383

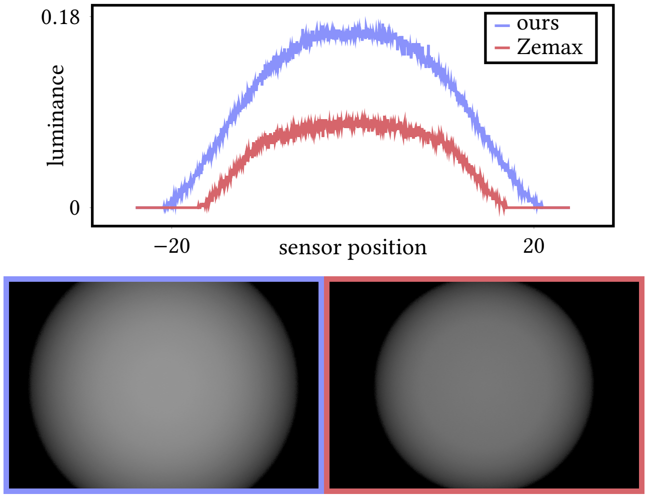
Ziyi Zhang, Nicolas Roussel, and Wenzel Jakob. 2023. Projective Sampling for Differentiable Rendering of Geometry. *ACM Trans. Graph.* 42, 6, Article 212 (dec 2023), 14 pages. https://doi.org/10.1145/3618385



high throughput weight (SNR = 46.85 dB)

low throughput weight (SNR = 35.86 dB)

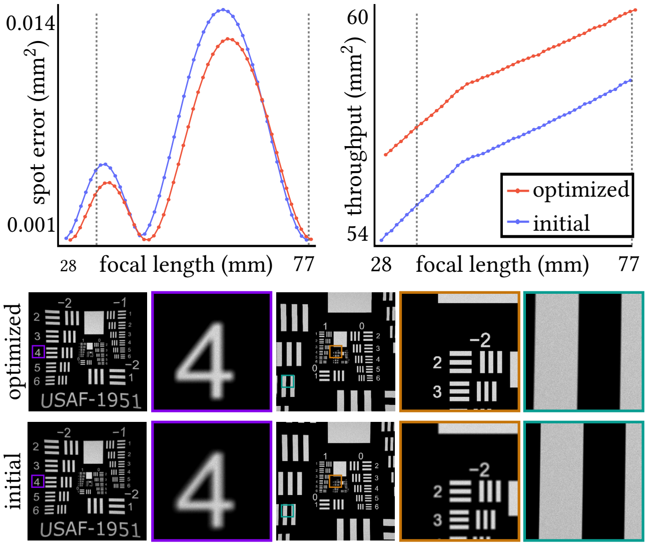
Figure 8: We simulate images of a candlelit scene using lenses optimized for high throughput ($\lambda = 5 \times 10^{-4}$, bottom left) versus small spot size ($\lambda = 1 \times 10^{-5}$, bottom right). We incorporate shot and read noise in rendered images—simulated for a Canon 7D sensor [Clark 2023] at unity ISO (984)—then gamma-correct ($\gamma = 0.2$) the noisy images. We also plot the histogram of pixel intensities in the two images (top). The image from the high-throughput lens has less pronounced noise and an intensity distribution shifted towards larger values; but also slightly worse blur (e.g., at the candle flame).



ours

Zemax

Figure 9: We simulate images of a white balance target using lenses optimized with our technique (bottom left) versus Zemax [2023] (bottom right). We also plot the horizontal cross section of the two images (top). By emphasizing throughput, including from off-axis sources, the lens from our technique strongly mitigates vignetting away from the sensor center.



35 mm

77 mm

Figure 10: We use our technique to optimize a zoom lens design in the focal length range 28 mm to 77 mm. We sum the composite objective of Equation (20) at three target focal lengths, 28 mm, 45 mm, and 77 mm. We plot the spot-size error (top left) and throughput (top right) of the initial and optimized designs across the focal length range, and simulate images of a USAF resolution target at focal lengths 35 mm (bottom left) and 77 mm (bottom right). The optimized design has overall better spot-size error and throughput across the focal length range. At the target focal length 77 mm, both lenses achieve the same overall spot-size error, but distribute the error to different parts of the sensor (insets).