Differentiable Compound Optics and Processing Pipeline Optimization for End-to-end Camera Design

ETHAN TSENG∗, Princeton University, United States
ALI MOSLEH∗ and FAHIM MANNAN∗, Algolux, Canada
KARL ST-ARNAUD, AVINASH SHARMA, and YIFAN PENG, Algolux, Canada
ALEXANDER BRAUN, Hochschule Dusseldorf, Germany
DEREK NOWROUZEZAHRAI, McGill University, Canada
JEAN-FRANÇOIS LALONDE, Université Laval, Canada
FELIX HEIDE, Princeton University, United States and Algolux, Canada

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery. 0730-0301/2021/8-ART0 $15.00 https://doi.org/10.1145/nnnnnn.nnnnnn

Most modern commodity imaging systems we use directly for photography—or indirectly rely on for downstream applications—employ optical systems of multiple lenses that must balance deviations from perfect optics, manufacturing constraints, tolerances, cost, and footprint. Although optical designs often have complex interactions with downstream image processing or analysis tasks, today’s compound optics are designed in isolation from these interactions. Existing optical design tools aim to minimize optical aberrations, i.e., deviations from Gauss’ linear model of optics, instead of application-specific losses, precluding joint optimization with hardware image signal processing (ISP) and highly-parameterized neural network processing. In this paper, we propose an optimization method for compound optics that lifts these limitations. We optimize entire lens systems jointly with hardware and software image processing pipelines, downstream neural network processing, and with application-specific end-to-end losses. To this end, we propose a learned, differentiable forward model for compound optics and an alternating proximal optimization method that handles function compositions with highly-varying parameter dimensions for optics, hardware ISP and neural nets. Our method integrates seamlessly atop existing optical design tools, such as Zemax. We can thus assess our method across many camera system designs and end-to-end applications. We validate our approach in an automotive camera optics setting—together with hardware ISP post-processing and detection—outperforming classical optics designs for automotive object detection and traffic light state detection. For human viewing tasks, we optimize optics and processing pipelines for dynamic outdoor scenarios and dynamic low-light imaging. We outperform existing compartmentalized design or fine-tuning methods qualitatively and quantitatively, across all domain-specific applications tested.

Fig. 1. We propose an end-to-end camera design scheme that jointly optimizes compound optics together with hardware and software image post-processors. Our approach allows us to cater lens systems and the hyperparameter settings of the entire imaging pipeline towards domain-specific applications, including but not limited to automotive object detection and natural image capture. We leverage existing traditional optics design tools such as Zemax, which can be easily integrated into our framework. We validate our method in simulations and in the real-world via manufactured prototypes, using an (optimized) ARM Mali-C71 ISP and cameras mounted onto an acquisition vehicle (left). When optimizing for perceptual image quality (center), the proposed method finds an optics and processing pipeline that improves visual detail across fields, outperforming conventional pipelines with a nominal lens resulting from over one month of Zemax-aided expert-design. When optimizing for a pedestrian-vehicle detection (right), the same method learns different optics with comparable spotsize but significantly higher speed (f/3.2) than the nominal optics (f/4.4), improving object detections in flux-limited image regions that are challenging to recover by fine-tuned conventional imaging and detection pipelines.

∗indicates equal contribution.
1 INTRODUCTION

Nearly all commodity camera systems rely on compound optics. These cascades of individual lens elements are designed to focus light reflected from scene surfaces onto sensor pixels. Combined with real-time processing in image signal processing (ISP) hardware, conventional camera systems are the foundation for ubiquitous applications in personal photography, communication, surveillance, and, with image data consumed by computer vision software, emerging applications in robotics and autonomous driving.

While processing algorithms and hardware have developed rapidly during the last decades, achieving impressive image reconstruction capabilities [Chen et al. 2018; Hasinoff et al. 2016], existing optical systems are still conceived in isolation. Indeed, optical systems are designed to reduce optical aberrations—i.e., deviations from linear optics [Gauss 1843]—independently of downstream tasks such as detecting objects for autonomous driving, or generating visually pleasing images for mobile photography. Existing optics optimization methods rely on a rich ecosystem of optimization tools [Garrard et al. 2005; Geary 2002], including 0th-, 1st-, and 2nd-order methods, but do not allow for joint optimization over the entire imaging pipeline—including the optics, ISP and all subsequent software processing—since the number of parameters to optimize becomes prohibitively large. Indeed, conventional compound optics, hardware ISPs and downstream convolutional neural network (CNN) architectures combine to form a complex, high-dimensional parameter space—with both categorical and continuous variables—with tens to millions of parameters, depending on the domain-specific application.

Recent end-to-end optics designs, using differentiable single-phase plate optics and software post-processing [Sitzmann et al. 2018], target this gap. These methods, however, require manufacturing complex photo-lithographic phase plates which are not applicable to commodity optical systems. Such systems are limited to a catalog of standard optical surfaces and a restricted number of ground, or injection-molded, aspherical elements. These constraints result from mass-market manufacturing processes and the high throughput required by real-time applications. As such, compound optics strike a careful balance between the availability of glasses, tolerance design in manufacturability, and compartmentalized manufacturing expertise perfected in the industry.

We propose the first approach to allow for joint optimization over the space of manufacturable compound optics, hardware or software ISPs, and downstream tasks. We propose an alternating proximal optimization method that combines 1st-order optimization for deep neural network detectors and other processing methods with large parameter spaces, non-differentiable compound lens parameters, and hardware ISP hyperparameters. Our method is the first to bridge traditional design with proprietary tools, such as Zemax, and stochastic 1st-order optimization. We can readily import existing Zemax designs and export to this industrial standard format. To this end, we build atop traditional tolerance design.

We validate our method across application-specific camera system designs and tasks. Specifically, we co-design automotive camera optics together with hardware ISP post-processing and object detection. We also optimize alternative optics and post-processing for human viewing, and for low-light imaging. Our approach qualitatively and quantitatively outperforms existing compartmentalized design or fine-tuning on every domain-specific application we tested.

Specifically, our work makes the following contributions:

• A joint end-to-end optimization framework for compound lens models combined with realistic sensor models, hardware (or learned software) image processing, and CNN computer vision modules. We jointly optimize the parameters and hyperparameters of this heterogeneous pipeline, using domain-specific losses.

• Our framework builds atop traditional tolerance analysis and integrates seamlessly with standard optics design methods.

• We combine existing raytracing algorithms with deep learning to construct accurate differentiable approximations of compound optics that are not limited to small fields of view within the paraxial regime.

• We analyze our method in simulation for optics and image processing tasks in human viewing and analytic downstream tasks.

• We build five real prototype lens designs based on our optimization framework, and validate the proposed approach on challenging real-world automotive data acquired on a test vehicle.

2 RELATED WORK

We review prior art most related to our methods and contributions.

Compound optics designs. While modern photography technologies have evolved to provide high-quality imaging performance across diverse devices and applications, the optical design process still follows classic aberration theory [Smith 2005] established over a century ago. Optical aberrations describe the deviations in focus—based on Gauss’ linear optics model [Gauss 1843]—resulting from optical path offsets of light as it travels across lens regions from varying incident directions [Fowles 1989]. In this way, stacks of several optical elements of varying surface shapes and materials are introduced to combat both monochromatic and chromatic aberration [Kingslake and Johnson 2009; Slusarev 1984]. This design methodology, when combined with the increases in sensor resolutions and shrinking form factors, results in commercial optic designs composed of many (i.e., dozens) spherical or aspherical elements.

State-of-the-art optics design software, such as Zemax or Code V [Garrard et al. 2005; Geary 2002; Walker 2008], can optimize surface profiles of refractive lenses. These tools use mid-level metrics—so-called merit functions—which aim to strike a compromise across many criteria [Malacara-Hernández and Malacara-Hernández 2016], such as spatially-varying point spread function (PSF) size over target wavelength bands. Their (proprietary) optimization methods scale...
only to a dozens of continuous optical parameters. Open design tools, such as LENS FACTORY [Sun et al. 2015], also admit these limitations and are not able to match commercial tools in their breadth of supported designs. These tools are not suited for joint optimization of neural networks and/or hardware ISPs with discrete and continuous parameters.

**Forward lens simulation models.** A variety of approaches in computer graphics have been proposed for accurately simulating the behavior of complex lens assemblies, including ray-based models [Kolb et al. 1995], spectral models [Steinert et al. 2011], and efficient Monte Carlo sampling [Hanika and Dachsbacher 2014]. Simulated lens models have been used to optimize optical inspection systems [Retzlaff et al. 2016], or to fit predicted PSFs to captures using first-order approximations [Shih et al. 2012]. These forward models are however not end-to-end differentiable, and are restricted to spherical lenses. Handling richer designs—including aspherical lenses—requires modeling scattering, flare, vignetting, and propagation properly, which is not differentiable and requires expensive raytracing operations [Walker 2008]. Researchers have demonstrated that deep learning can closely approximate complex functions such as ISPs [Tseng et al. 2019] and physical simulators [Grzeszczuk et al. 1998], however, it is non-trivial to directly learn an accurate image-to-image model of compound optics that allows for gradient feedback as the PSFs are spatially varying and encode differences in the optical design parameters through subtle changes. We propose to combine deep learning with existing raytracing algorithms to create a differentiable module that accurately reproduces the behavior of complex lenses as a function of the optical parameters.

**Merit functions.** Existing optical designs are optimized for intermediate merit functions, such as the PSF size or wavefront errors [Smith 2005]. While these traditional design guidelines have led to high-quality camera lens designs, current merit functions are blind to downstream operations in the camera image processing pipeline. While the goal of a camera is to record a “perfect”—typically widely used in existing optical design works [Chang et al. 2018; Sitzmann et al. 2018], or to fit predicted PSFs to captures using first-order approximations [Shih et al. 2012]. These forward models are however not end-to-end differentiable, and are restricted to spherical lenses. Handling richer designs—including aspherical lenses—requires modeling scattering, flare, vignetting, and propagation properly, which is not differentiable and requires expensive raytracing operations [Walker 2008]. Researchers have demonstrated that deep learning can closely approximate complex functions such as ISPs [Tseng et al. 2019] and physical simulators [Grzeszczuk et al. 1998], however, it is non-trivial to directly learn an accurate image-to-image model of compound optics that allows for gradient feedback as the PSFs are spatially varying and encode differences in the optical design parameters through subtle changes. We propose to combine deep learning with existing raytracing algorithms to create a differentiable module that accurately reproduces the behavior of complex lenses as a function of the optical parameters.

**Image processing pipeline design.** Direct sensor measurements suffer from many sources of degradation including, but not limited to, the aforementioned aberrations, color filter subsampling, photon shot noise, cross-talk effect, and read-out noise. Custom optimized hardware ASICs are used—in part due to the performance critical nature of real-time systems built atop these imaging modules—to realize the low-level image processing needed to reconstruct high-quality images from the measurement polluted by these degradations [Hegarty et al. 2014; ON Semi MT9P001 2017; Ramanath et al. 2005; Shao et al. 2014; Zhang et al. 2011].

Any ISP designs that deviate from this model are typically limited, in their deployment, to off-line tasks. Optimization-based methods [Heide et al. 2014] operate orders of magnitude slower than real-time ISPs. Machine learning-based methods focus on specific tasks, such as demosaicking [Gharbi et al. 2016], tonemapping [Gharbi et al. 2017], low-light denoising [Chen et al. 2018] and other common processing operators [Chen et al. 2017; Fan et al. 2018; Xu et al. 2015]. These methods require high-end GPUs with high power consumption (i.e., ≥ 100 Watts). Despite this, such data-driven approaches remain attractive in their ability to be tailored to specific end tasks, but they remain limited to simpler single-element optical designs. We will demonstrate joint optimization of both hardware and software ISP with complex multi-element optics.

Recent approaches that explore camera pipeline optimization cannot jointly optimize the ISP with downstream applications [Nishimura et al. 2018], are limited to optimizing individual differentiable blocks [Li et al. 2018], or tackle end-to-end ISP optimization without any consideration for the optics [Tseng et al. 2019]. While Li et al. [2018] does provide a differentiable optical model using raytracing, their optical design strategy entirely relies on intermediary heuristics without consideration of the final endpoint loss. Mosleh et al. [2020] recently proposed an ISP optimization method using hardware-in-the-loop, however, this approach is currently impractical for optics design as new lenses would need to be manufactured and tested at each iteration of the optimization.

**End-to-end optical design.** Recent works explore the applicability of diffractive optical elements [Chang et al. 2018; Metzler et al. 2020; Sitzmann et al. 2018; Stork and Gill 2014; Sun et al. 2020] for imaging in photography and other vision-based applications, here still with a single optical element. This single element restriction, coupled with an approximate forward Fresnel propagation model, is needed to render their joint automated design optimization task tractable. These simplifications, however, come at a cost: the realizable diameter and field of view of the resulting lens is limited, and resulting imaging quality lags behind that of commodity multi-element compound lens designs. Moreover, the resulting designs are not suitable for production systems, as no tolerance analysis is considered. Peng et al. [2019] address some of these limitation with a single, hand-crafted free-form element. This hand-tuned design does not support joint optimization with image processing hardware and software, nor does it support multi-element compound lenses. Our work lifts these important remaining limitations, incorporating traditional tolerance analysis to design compact optics in an end-to-end fashion. Parallel work from Sun et al. [2021] addresses compound lens optimization. While their work relies on differentiable ray-tracing, the proposed method does not require a differentiable forward model, allowing us to integrate our method with existing lens design tools such as ZEMAX.

3 IMAGING PIPELINE STAGES

We present our overarching image formation model and imaging pipeline stages. Our imaging pipeline is divided into five (5) core stages (Fig. 2): the scene, compound camera optics, sensor, ISP (hardware or software), and downstream tasks. We detail each, below.

3.1 Scene representation

We treat scenes as all-in-focus RGB images and assume that all scene content lies beyond the hyperfocal distance, a representation widely used in existing optical design works [Chang et al. 2018; Sitzmann et al. 2018]. It is similarly suitable here, since we target

ACM Transactions on Graphics, Vol. 40, No. 0, Article 0. Publication date: August 2021.
fixed focus (e.g., automotive imagers for object detection [Geiger et al. 2013]) rather than depth-sensitive applications (e.g., mobile photography) for which auto-focus is necessary. Treating the entire scene as being in focus also allows us to downsample scene data to suit downstream hardware and software ISPs, providing scene-scale invariance that decouples acquisition optics from image post-processing. RGB images allow us to build upon existing RGB training data and established methods for RGB image processing, while being computationally less costly than, e.g., computing multispectral PSFs.

3.2 Compound camera optics

Existing differentiable optical design approaches [Chang and Wetzelstein 2019; Sitzmann et al. 2018] rely on the paraxial approximation, reducing the optical response to a single PSF and enabling a compact differentiable Fourier propagation model based on wave optics. This approximation, however, only holds for small fields of view (FOV $\approx 5^\circ$), whereas full ray-tracing is required to accurately model optics for larger FOV. Moreover, these methods only design a single optical element, whereas consumer and industrial optical systems commonly consist of a sequence of many such elements.

Our work develops a framework for the design of such compound optical systems, that is, optical systems consisting of multiple optical elements. Our model is not limited to the paraxial regime, and so, can handle wide FOVs. We achieve this by simulating spatially-varying PSFs that describe the features produced by complex optical pipelines – including Seidel aberrations and vignetting – which cannot be described using a single, spatially-invariant PSF.

We parameterize an optical system (with a fixed number of lens elements) by the set $\mathcal{P}_{\text{optic}}$, which includes surface thicknesses $t$, intervals $l$, refractive indices $\eta$, and surface parameters $s$ for every element, as well as the stop position. We assume the f-number and back-focal length of the optical system are given as fixed design constraints during optical design. Fig. 3 illustrates a three-element system; we will introduce several alternatives in Sec. 6.

Assuming scene content that lies at infinity (Sec. 3.1), the spherical light rays from a scene surface point enter the pupil of the optical system in parallel at angle $\Theta$. An ideal optical system transfers a wavefront at angle $\Theta$, i.e., the position of equal phase, to a perfect inverted spherical wave centered at the image plane (polar) coordinate $r$. In this ideal system and assuming a rotationally symmetric optical system, sources at infinity produce images at $r = F \tan \Theta$ with $F$ as the focal distance to the image plane. Deviations from this ideal behavior are typically measured as an optical path difference (OPD) between the ideal and the system wavefronts, expressed as a function $f_{\text{psf}}(p, r, \lambda; \mathcal{P}_{\text{optic}})$ of the exit pupil plane position $p$, image coordinate $r$, and wavelength $\lambda$, for optics parameters $\mathcal{P}_{\text{optic}}$.

We model the spatially-varying PSF response of a compound optical system as the following function $f_{\text{psf}}$:

$$
\text{PSF}_\lambda(x, r; \mathcal{P}_{\text{optic}}) = \left| \int A(p) e^{i f_{\text{psf}}(p, r, \lambda; \mathcal{P}_{\text{optic}})} e^{i 2\pi px} dp \right|^2
$$

(1)

where $x$ is the spatial coordinates in the PSF. The spatially-varying PSF for a given radial position $r$ is thus the magnitude of the inverse Fourier transform over the exit pupil. We assume the amplitude of the exit pupil to be unattenuated, $A(\cdot) = 1$. Note that traditionally the OPD is optimized by lens designers where the polynomials for $r$ and $p$ are evaluated, such as the 3rd-order Seidel aberrations. So-called merit functions are also placed on these individual aberrations.
Photons are then converted into electrons using the detector quantum efficiency \( \eta(x, y, \lambda) = \epsilon(x, y, \lambda) / \rho(x, y, \lambda) \), where \( \epsilon(x, y, \lambda) \) is the number of electrons generated when \( \rho(x, y, \lambda) \) photons arrive at a sensor location \((x, y)\) for wavelength \(\lambda\).

Other factors also lead to electron generation, such as temperature and electronic imperfections. Our sensor model includes dark noise \( n_d \sim N(\mu_d, \sigma_d)\) (electron noise generated in the absence of light) as well as dark current \( \eta_t \) (electron noise dependent on the sensor temperature \( T \)), which follows a Poisson distribution with mean

\[
\mu_t = \mu_{t,\text{ref}} \cdot \sqrt{(T - T_{\text{ref}}) / T_d} \cdot \text{texp}.
\]

Here, \( \mu_{t,\text{ref}} \) is the average dark current measured at a reference temperature \( T_{\text{ref}} \), \( T_d \) is the temperature interval that causes a doubling of the dark current, and \( \text{texp} \) is the exposure time.

Finally, we convert electrons to digital values by clipping electron quantities at the maximum well capacity \( \epsilon_{\text{sat}} \), and scaling by a gain factor \( K \), before quantizing and adding black level \( B_n \). Note that clipping is commonplace in machine learning, e.g., with ReLU activations. To permit differentiability of the quantization step, we simulate quantization with uniform noise \( n_q \sim \text{Uniform}(0, 0.5, 0.5)\). Thus, the digital readout \( I_{\text{raw}} \) at position \((x, y)\) when \( \rho(x, y, \lambda) \) photons arrive at the sensor is given by

\[
I_{\text{raw}}(x, y) = b + n_q + K \min\left(\epsilon_{\text{sat}}, n_q + \eta_t \sum_{\lambda} \rho(x, y, \lambda) \eta(x, y, \lambda)\right). \tag{5}
\]

### 3.4 Hardware imaging pipeline stages

Hardware ISPs, such as the ARM Mali C71, are becoming increasingly ubiquitous due to their real-time performance, power efficiency, and high resolution, all of which are critical to dynamic applications such as automotive imaging. ISPs operate directly on RAW images captured on camera sensors and, after a series of individual image processing stages, they output an image ready for human consumption or for further downstream processing. We describe these individual processing blocks below:

1. **Color-correction, gain**: Variations in quantum efficiencies cause CFA filters to treat wavelengths non-uniformly. As such, color-correction (e.g., white-balance) and gain-adjustment are often applied after black level removal [Ramanath et al. 2005].

2. **Demosaicking**: Sensor detectors are commonly arranged in an alternating R-G-G-B Bayer mosaic pattern. A demosaicking stage reconstructs missing color information at each pixel to produce trichromatic RGB images. Bilinear interpolation between neighboring pixels is a common strategy, here [Zhang et al. 2011].

3. **Denoising**: Noise can be reduced using methods like edge-preserving filters [Choi et al. 2014; Tomasi and Manduchi 1998] or non-local patch matching [Dabov et al. 2007; Zhang et al. 2016].

4. **Color, tone correction**: Image adjustments can be performed to improve overall appearance. These include both global (e.g., gamma correction) and local operations (e.g., edge sharpening).

5. **Color space conversion, compression, further processing**: Finally, the image can be converted into an output colorspace (e.g., sRGB or HSV), compressed (e.g., to jpeg) for transfer or storage, or further processed by downstream image processing applications or pipelines (e.g., object detection).
3.5 Software image processing and analysis

Much of modern image processing and computer vision is performed in software, allowing for flexible algorithm design. There are many software ISPs, e.g., bilateral filtering, non-local patch denoising, and deep neural image processing approaches. The latter have been applied to a broad range of imaging tasks, including demosaicking [Gharbi et al. 2016], denoising [Chen et al. 2018], and tone-mapping [Gharbi et al. 2017]. Recent work has also demonstrated the ability of deep neural networks to replicate existing ISP pipelines [Chen et al. 2017; Fan et al. 2018; Tseng et al. 2019; Xu et al. 2015].

Furthermore, they are increasingly used for post-capture down-stream tasks, such as face filtering, scene understanding, reconstruction, and object detection. Indeed, state-of-the-art performance for these vision tasks has been achieved with deep neural networks.

4 COMPOUND OPTICS PIPELINE OPTIMIZATION

We detail an end-to-end, differentiable model that implements each step of our image formation model (Sec. 3): compound optics, sensing, low-level and high-level image processing. We use our pipeline to jointly optimize optical design parameters $P_{\text{OPTIC}}$ and image post-processing parameters, which can include—but are not limited to—hardware ISP parameters $P_{\text{ISP}}$ and/or neural network weights $P_{\text{NN}}$ (see Fig. 2), in an end-to-end fashion.

4.1 Compound optics modeling

Although optics design software, such as Zemax and Code V, incorporate powerful optical simulators, their non-differentiable black-box nature prevents us from directly incorporating them into our end-to-end differentiable pipeline. We circumvent the non-differentiability of these systems by modeling them with a neural network $f_{\text{optic}}$ that we train to predict spatially-varying PSFs and vignette given optics parameters $P_{\text{OPTIC}}$. Specifically, from Eq. (1), we approximate the true optical function $f_{\text{optic}}$ with a network $f_{\text{optic}}$ parameterized by weights $W_{\text{optic}}$.

$$f_{\text{optic}}(r, P_{\text{OPTIC}}, \hat{W}_{\text{optic}}).$$

where $\hat{\phi}(r)$ and $\hat{\upsilon}(r)$ are the (estimated) PSF and vignette factors at field $r$. The neural network $f_{\text{optic}}$, illustrated in Fig. 5, separately outputs an energy-preserving PSF (unit sum across RGB channels in PSF$_{\text{est}}$) for a given radial position $r$ and a 3-vector (RGB) vignette. Finally, we scale the PSF channel-wise by the vignetting factor. We observe that separately predicting the normalized PSFs and vignette factors increases performance compared to direct prediction.

As shown in Fig. 5, our neural network architecture comprises a multi-layer perceptron (MLP) encoder combined with a convolutional decoder. We output the (per-PSF) vignette factor directly from the MLP, while the decoder generates the PSF. We obtain optical network weights $\hat{W}_{\text{optic}}$ by minimizing a loss $L_{\text{optic}}$ between the network predictions and ground truth $O$:

$$\hat{W}_{\text{optic}} = \arg \min_{W_{\text{optic}}} \sum_{j=1}^{M, K} \sum_{l=1}^{L_{j}} L_{\text{optic}}(f_{\text{optic}}(r^{(l)}), \phi^{(l)}(r^{(l)}), \upsilon^{(l)}), O^{(l,j)}).$$

Here, the sum is over $M$ optical designs, each with $K$ PSFs. We center and normalize optical parameters by their mean and standard deviation (across the $M$ optical designs) before passing them to $f_{\text{optic}}$. In our experiments, we uniformly sample spatial distances $r$ across $K = 13$ locations along the vertical axis of the input image and use $M = 5.5 \times 10^4$ designs. We opt for a discrete sampling instead of a dense continuous sampling of every pixel for computational efficiency; see supplemental document for details. The image $I_{\text{optic}}$ is reconstructed by rotating each $\phi(r)$ to obtain PSF predictions for all locations at distance $r$ from the center of the scene, and using the procedure from Sec. 3.2 (see Eq. (2)).

Our optics meta-network requires that the cardinality of input parameters be fixed. Lifting this requirement is a direction of future research. Nevertheless, training a suite of optics meta-networks for different parameter sets is feasible as the time for network training is around 6 hours. Note that robustness to manufacturing errors can be incorporated by adding noise to the input parameters, however, we did not find this necessary.

Training data generation. We obtain ground truth PSFs with traditional optics design software, e.g., OpticStudio by Zemax. Zemax allows us to accurately compute optical path differences with a time-consuming ray-tracer [Hanika and Dachsberger 2014; Harvey et al. 2015; Schrade et al. 2016; Steinert et al. 2011], including aspherical surfaces, scattering, flare and diffraction. From a basic lens system design (e.g., a 3-element design; Fig. 3), we randomly sample within predefined ranges for each parameter to generate the superset of $M$ optical parameters $P_{\text{OPTIC}}$ in Eq. (7).

Of note, however, is that we face additional constraints when determining plausible ranges for each parameter. Slight changes in parameters can greatly affect the performance of the optical system, and we are additionally bound by constraints imposed by the manufacturing process as we wish our lens designs to be physically realizable. We perform a tolerance sensitivity analysis in Zemax.
to enforce this. Once we assign a viable tolerance to each component, and subsequently determine parameter ranges, we generate thousands of random variations of the compound lens.

For each lens variation we uniformly sample FOVs, and we obtain their corresponding PSFs (projected onto the sensor plane) with ray-tracing and PSF simulation in Zemax. Since we assume rotational symmetry in the compound lens designs (Sec. 3.2), we only simulate PSFs sampled from the positive vertical axis of the FOV. Also, during PSF simulation, our sampling accounts for the target sensor resolution. In practice, training the optics PSF representation model with super-resolved PSF data leads to more accurate fits: e.g., in Fig. 3, we sample the target FOV uniformly, and we simulate the corresponding PSFs for 128 × 128 micrometer sensor areas while the target sensor pixel size is 5.86 μm. For PSF simulation, we rely on Huygens PSF calculation of the optical system [Sun 2016], which contains 7 optical elements. We measured PSFs with a Kowa 1/2” LM6NCL lens [Kowa 2020], which contains 7 optical elements. We measured PSFs with a Trioptics ImageMaster using broad spectrum and a “photopic eye”

**Loss function.** We compute the loss \( L_{\text{optic}} \) from Eq. 7 on the estimated (energy-preserving) PSF \( \hat{\psi} \) and vignette factor \( \hat{v} \) as

\[
L_{\text{optic}}(\hat{\psi}, \hat{v}, \psi^*, v^*) = L_1(\hat{\psi}, \psi^*) + L_1(\hat{v}, \psi^*) + \sum_d L_1(\nabla_d \hat{\psi}, \nabla_d \psi^*) + L_1(\hat{v}, v^*),
\]

(8)

where \( F(\cdot) \) is the Fourier transform, and \( \nabla_d \) is the forward difference operator in direction \( d \). Here, superscripts \( ^* \) indicate ground truth values. Please refer to our supplemental document for more details on network architecture, training, and datasets.

**Validating the optics meta-network.** Before detailing the remaining components of our method, we illustrate the capabilities of our optics network \( \hat{f}_{\text{optic}} \) in Fig. 6 for accurately parameterizing PSFs on a real compound lens—here the Kowa 1/2” LM6NCL lens [Kowa 2020], which contains 7 optical elements. We measured PSFs with a Trioptics ImageMaster using broad spectrum and a “photopic eye” filter, which transmits light in proportion to the human eye’s natural response [Galvoptics 2020]. We measured PSFs at five distances from the image plane, \( \pm (0, 20, 40) \) mm and train \( \hat{f}_{\text{optic}} \) to reproduce these measured PSFs given the distance to the image plane. The qualitative results in Fig. 6 demonstrate that our optics network can accurately reproduce spatially-varying PSFs, even with minute details encoded depending on the design parameters. We provide further validation of the optics network in our supplemental document.

### 4.2 Differentiable sensor and ISP model

We implement our differentiable sensor model \( \hat{f}_{\text{sensor}} \) as described in Sec. 3.3: it accepts the post-optic image \( I_{\text{optic}} \) as input and outputs the sensor-produced RAW image \( I_{\text{raw}} = \hat{f}_{\text{sensor}}(I_{\text{optic}}) \). We feed this RAW image into the post-processing ISPs. As mentioned in Sec. 3.4, our model supports both hardware and software ISPs.

**Hardware ISPs.** We simulate hardware ISPs (Sec. 3.4) using the approach of [Tseng et al. 2019], who learn to approximate the behavior of an ISP using a deep UNet-style CNN. Similarly, we use a network to learn the mapping from an input RAW image \( I_{\text{raw}} \) and ISP parameters \( \mathcal{P}_{\text{isp}} \) to the ISP output image \( I_{\text{isp}} = \hat{f}_{\text{isp}}(I_{\text{raw}}, \mathcal{P}_{\text{isp}}; W_{\text{isp}}) \), where \( W_{\text{isp}} \) are trainable network weights obtained by minimizing

\[
W_{\text{isp}} = \arg \min \{ W_{\text{isp}} \} \sum_I L_{\text{isp}}(\hat{f}_{\text{isp}}(I, \mathcal{P}_{\text{isp}}; W_{\text{isp}}), O) \quad (9)
\]

on a set of \( M \) input/output \( (I, O) \) training pairs [see [Tseng et al. 2019]]. We combine an \( L_1 \) loss on the image domain and a perceptual loss (from a pre-trained AlexNet [Zhang et al. 2018]).

We base the network architecture for \( \hat{f}_{\text{isp}} \) on a UNet, which accepts a multi-channel tensor as input, with the input RAW image \( I_{\text{raw}} \) as the first channel, and the remaining channels are the ISP parameters (with each parameter replicated to fill an entire channel). This mirrors [Tseng et al. 2019], with the exception that we prepend a non-trainable bilinear demosaicking layer to the network to handle varying CFA patterns. This layer converts the single channel RAW sensor image into an RGB tensor. Note that we are not limited to bilinear demosaicking and that any differentiable demosaicker can be used for this step. The trained ISP proxy is appended to the remainder of the pipeline, obtaining \( I_{\text{isp}} = \hat{f}_{\text{isp}}(I_{\text{raw}}, \mathcal{P}_{\text{isp}}; W_{\text{isp}}) \).

**Software ISPs.** Software ISPs (Sec. 3.5), particularly those parameterized as deep neural networks, are trivial to employ in our pipeline in a differentiable manner. We similarly append software ISPs \( f_{\text{isp}} \) parameterized by \( \mathcal{P}_{\text{isp}} \) to our pipeline as \( I_{\text{isp}} = f_{\text{isp}}(I_{\text{raw}}, \mathcal{P}_{\text{isp}}) \).

Note that here, we do not employ superscripts \( ^* \) as the network is not used to approximate a physical process (as \( \hat{f}_{\text{optic}} \) and \( \hat{f}_{\text{isp}} \) approximated both physical optics and ISP respectively).

### 5 Joint Optimization

#### 5.1 Fully differentiable imaging pipeline

Our full end-to-end pipeline using a hardware ISP is given as

\[
O = \hat{f}_{\text{isp}}(f_{\text{sensor}}(\hat{f}_{\text{optic}}(I, \mathcal{P}_{\text{optic}}; W_{\text{optic}}), \mathcal{P}_{\text{isp}}; W_{\text{isp}}), \mathcal{P}_{\text{isp}}; W_{\text{isp}}) \quad (10)
\]

where \( I \) is the input RGB image, \( O \) is the output image, and \( \hat{f}_{\text{optic}} \) produces the post-optic image as described in Sec. 4.1.

If instead employ a software ISP then our full pipeline becomes

\[
O = f_{\text{isp}}(f_{\text{sensor}}(\hat{f}_{\text{optic}}(I, \mathcal{P}_{\text{optic}}; W_{\text{optic}}), \mathcal{P}_{\text{isp}}; W_{\text{isp}}), \mathcal{P}_{\text{isp}}; \mathcal{P}_{\text{isp}}) \quad (11)
\]

Due to the differentiability of each of pipeline stage, we can concatenate arbitrarily many image post-processing stages. One such example is for automotive object detection, where the sensor RAW image is first processed by a hardware ISP before being fed into an object detection network. In this case, our full pipeline is

\[
O = f_{\text{isp}}(f_{\text{sensor}}(\hat{f}_{\text{optic}}(I, \mathcal{P}_{\text{optic}}; W_{\text{optic}}), \mathcal{P}_{\text{isp}}; \mathcal{P}_{\text{isp}}), \mathcal{P}_{\text{isp}}; \mathcal{P}_{\text{isp}}) \quad (12)
\]

At this point, \( \hat{f}_{\text{optic}} \) and \( f_{\text{isp}} \) are fully trained, and so their weights \( \{W_{\text{optic}}, W_{\text{isp}}\} \) are fixed (but included in Eq. (12) for completeness).

We can now minimize task-specific losses \( L_{\text{task}} \) with respect to the system parameters \( \{\mathcal{P}_{\text{optic}}, \mathcal{P}_{\text{isp}}, \mathcal{P}_{\text{isp}}\} \) in order to determine what the best combination of optics, ISP and image processing parameters are in a task-dependent manner:

\[
\{\mathcal{P}_{\text{optic}}, \mathcal{P}_{\text{isp}}, \mathcal{P}_{\text{isp}}\} = \arg \min \{ \mathcal{P}_{\text{optic}}, \mathcal{P}_{\text{isp}}, \mathcal{P}_{\text{isp}} \} \sum_{i=1}^{M} L_{\text{task}}(O(i), T(i)) \quad (13)
\]

Here, example tasks include image-to-image translation, where the target T is a desired high-quality image, or an image-to-abstraction task with scene segmentation or a bounding box map targets.
5.2 Proximal Compositional Optimization

Joint end-to-end optimization of several different processing blocks is challenging due to many local minima in the loss landscape and sensitivity to initialization states. We detail an optimization method that allows for efficient end-to-end design below. Note that while we do not provide formal guarantees, we validate the method extensively in Sec. 6.3.

Our method optimizes differentiable compositions of functions, $F(x, \cup_{i=1}^{K} \mathcal{P}_i) = f_K(f_{K-1}(\ldots f_1(x, \mathcal{P}_1) \ldots, \mathcal{P}_{K-1}), \mathcal{P}_K)$, (14) with respect to a global loss $L$, where differentiable functions $f_i$ depend on parameters $\mathcal{P}_i$, Eq. (12) is one instance of this class of functions, where $x$ is an RGB image, $f_1 = f_{\text{optic}}, f_2 = f_{\text{sensor}}, f_3 = f_{\text{isp}}, f_4 = f_{\text{sensor}}$ and $\mathcal{P}_1 = \mathcal{P}_{\text{optic}}, \mathcal{P}_2 = 0, \mathcal{P}_3 = \mathcal{P}_{\text{isp}}, \mathcal{P}_4 = \mathcal{P}_{\text{sensor}}$. Our algorithm (see listing Alg. 1) operates as follows.

**Initialization.** Careful parameter initialization is key to exploring a diversity of possibilities, while avoiding local minima. Each $\mathcal{P}_i$ can be initialized through random sampling or from a pre-determined initialization. In our experiments, we initialize $\mathcal{P}_{\text{optic}}$ with uniform random sampling. We always initialize ISP parameters $\mathcal{P}_{\text{isp}}$ randomly and uniformly. Network software ISPs consist of many more parameters than the other stages and so, to avoid local minima, we first pre-train the software ISPs on synthetic training data.

**Compositional optimization.** We individually optimize each parameter set $\mathcal{P}_i$ for $n_i$ steps in a round-robin fashion. All parameters are optimized with respect to the same task loss $L_{\text{task}}$. We train for $n_c$ cycles, where each cycle consists of $\sum_{i=1}^{K} n_i$ training steps. This alternating optimization scheme yields finer control over the optimization of individual function blocks.

**Proximal regularization.** If a certain stage evolves too rapidly, then it may be difficult to optimize the other stages in tandem. We propose and employ a proximal regularizer (inspired by Eq. (1.3b) of [Xu and Yin 2013]) to stabilize training. Specifically, our proximal regularization loss for a specific parameter vector $\mathcal{P}_i$ is

$$L_{\text{prox}}(\mathcal{P}_i(t), \beta_i) = \beta_i \| \mathcal{P}_i(t) - \mathcal{P}_i(t+1) \|_2^2,$$

(15)

where $\mathcal{P}_i(t)$ and $\mathcal{P}_i(t+1)$ are the current and next iterates, and $\beta_i$ is a scalar weight. During compositional optimization we add $L_{\text{prox}}$ to the global task loss $L_{\text{task}}$.

**Fine-tuning.** After $n_c$ cycles, we train all parameters end-to-end—without alternating nor proximal regularization—for $n_e$ cycles.

6 ANALYSIS AND SYNTHETIC VALIDATION

In this section, we first validate the utility and effectiveness of the proposed method using simulated optical designs.

6.1 Cooke triplet optimization

We first present a series of experiments that employ our optics network $f_{\text{optic}}$ to determine the optimal task-specific parameters of a “Cooke triplet” [Kidger 2002], an established lens design composed of three optical elements which corrects Seidel aberrations. To minimize manufacturing efforts for the handful of lens systems
Differentiable Compound Optics and Processing Pipeline Optimization for End-to-end Camera Design

ALGORITHM 1: Proximal Compositional Optimization

Initialize()
for $t = 1, \ldots, n_f$ do
  // Compositional Optimization
  for $f_i \in \{f_1, \ldots, f_K\}$ do
    // Round-robin over $f_i$
    for $j = 1, \ldots, n_i$ do
      $L' = L_{task} + L_{prox}(P_i, \beta_i)$
      Update $\{P_i, \partial L / \partial P_i\}$
    end
  end
end
for $t = 1, \ldots, n$ do
  // Fine-tuning
  Update $\{\bigcup_{i=1}^{K} P_i, \partial L / \partial \bigcup_{i=1}^{K} P_i\}$
end

that we intended to fabricated for this work, we select an off-the-shelf bi-concave glass element (Thorlabs LD2297-A in BK7 material) as the center element. Not only does this allow us to employ two material types in our design (as our manufacturing facilities were limited to PMMA lens fabrication), it has the added benefit that this element is coated with an anti-reflective film, reducing lens flare. All experiments use the methodology presented in Sec. 4.1 for training $\tilde{f}_{optic}$, but each adapt the loss function $L_{task}$ from Eq. (13) and training set to different tasks. For our sensor simulation, we have calibrated a 2.3 megapixel Sony IMX249 sensor with IR cutoff filter (specifications BFLY-U3-23S6C-C) with exposure of 5 ms, see supplemental document for details.

For all experiments, the optics parameters to optimize $P_{optic}$ are the ones shown in Tab. 1, where each lens element is denoted by its two surfaces. We consider the first and the sixth surfaces as rotationally symmetric polynomial aspheric surfaces, and optimize their spherical radius parameters and higher-order aspheric coefficients. While these parameters are all continuous, discrete parameters can be handled using the approach of Tseng et al. [2019] by using a continuous relaxation.

Nominal Optics Design. The experiments below compare optimized optical parameters against a “nominal” design, obtained using the Hammer optimization in ZEMAX with the primary goal of enforcing as focusing performance as much as possible across the field of view. To this end, we apply the default OPD (optical path difference) merit function with the following physical constraints: effective focal length, element thickness, air gap, and back focal distance. These constraints still permit a high degree of freedom, from which the Hammer optimization yields a high quality optic.
Fig. 8. Image quality with Cooke triplet and hardware ISP using simulated measurements. Although both PSFs have a support (number of non-zero entries on sensor) similar in size, the long streak component of the nominal optics (left) results in the ISP tends to overly unsharp mask which generates shadow artifacts and noise. The optical images produced using our optimized optics (right) exhibit a more circular distribution and the ISP output hence has less artifacts. Displayed optics PSFs have been resampled for the sensor array. We show the PSFs along the first half of the main diagonal of the 1200 × 1920 sensor. The center pixel coordinates (row, column) of the spatial PSFs are indicated above each PSF, where (0, 0) refers to the top-left corner.

Image quality with software ISP. We begin with the common task of capturing images for human viewing. Here, we use a software ISP \( f_{\text{isp}} \) (Eq. (11)) to perform the tone mapping operation immediately after the sensor readout. A standard UNet architecture is used for \( f_{\text{isp}} \). The training set is the MIT-Adobe FiveK dataset [Bychkovsky et al. 2011] and the RGB scenes used are under good-lighting conditions with high photon flux. The inputs are linear RGB images and the target tone-mapped images follow the tone-mapping performed by expert tuning. The simulated imaging pipeline from Eq. (11) is trained to produce the target tone-mapped images as closely as possible using a weighted perceptual quality loss. Specifically, we use \( L_{\text{task}} = L_1 + L_{\text{LPIPS}} \) where \( L_{\text{LPIPS}} \) is the perceptual loss using pre-trained AlexNet described by Zhang et al. [2018]. To simultaneously optimize for \( P_{\text{optic}} \) and \( P_{\text{nn}} \), we employ the alternating scheme from Sec. 5.2. For fair comparison against the nominal optic, we optimize \( f_{\text{isp}} \) with the same task loss while keeping \( f_{\text{optic}} \) fixed at the nominal parameter settings.

Through our end-to-end optimization process, we observe that the optimized optic design sacrifices the sharp focus at the center of the FOV in return for tight PSFs that minimize chromatic aberrations across the sensor FOV, see PSFs in Fig. 7. These traits enable superior joint performance with the software ISP as can be seen qualitatively for the images (post processing) in Fig. 7 and quantitatively in Tab. 2. Please see the supplemental document for additional results.

Image quality with hardware ISP. We perform the same experiment using a hardware ISP \( \tilde{f}_{\text{isp}} \) as a post-processor instead of the neural network \( f_{\text{isp}} \). Here, we use the ARM Mali-C71 hardware ISP and train \( \tilde{f}_{\text{isp}} \) as in Sec. 4.2. We again employ alternating optimization from above to minimize \( L_{\text{task}} = L_1 + 3L_{\text{LPIPS}} \). We see that the
Table 1. Parameters for the three-element Cooke triplet. We follow the optics CAD terminology and denote each lens element by its two surfaces [Garrard et al. 2005]. Accordingly, we refer to the aperture and the imaging plane by surface 5 and surface 8, respectively. We enforce the min/max constraints for all optimized lenses and the nominal lens.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1_radius</td>
<td>9.45</td>
<td>14.98</td>
<td>mm</td>
<td>radius of the 1st surface</td>
</tr>
<tr>
<td>s_1_conic</td>
<td>-0.49</td>
<td>0.29</td>
<td>-</td>
<td>conic constant of the 1st surface</td>
</tr>
<tr>
<td>s_2_radius</td>
<td>9.45</td>
<td>14.82</td>
<td>mm</td>
<td>radius of the 2nd surface</td>
</tr>
<tr>
<td>l_12</td>
<td>4.58</td>
<td>10.11</td>
<td>mm</td>
<td>distance between lens 1 and lens 2</td>
</tr>
<tr>
<td>l_2STO</td>
<td>1.03</td>
<td>9.22</td>
<td>mm</td>
<td>distance between lens 2 and aperture</td>
</tr>
<tr>
<td>s_6_conic</td>
<td>-0.49</td>
<td>0.49</td>
<td>-</td>
<td>conic constant of the 6th surface</td>
</tr>
<tr>
<td>s_6_radius</td>
<td>13.38</td>
<td>18.17</td>
<td>mm</td>
<td>radius of the 6th surface</td>
</tr>
<tr>
<td>s_6_10th</td>
<td>-1.28e-10</td>
<td>1.08e-10</td>
<td>mm</td>
<td>10th order coefficient of polynomial fit to 1st surface</td>
</tr>
<tr>
<td>s_1_10th</td>
<td>-1.28e-10</td>
<td>1.08e-10</td>
<td>mm</td>
<td>10th order coefficient of polynomial fit to 1st surface</td>
</tr>
<tr>
<td>s_1_4th</td>
<td>-9.06e-5</td>
<td>-1.45e-5</td>
<td>mm</td>
<td>4th order coefficient of polynomial fit to 1st surface</td>
</tr>
<tr>
<td>s_6_6th</td>
<td>-1.44e-6</td>
<td>1.73e-6</td>
<td>mm</td>
<td>6th order coefficient of polynomial fit to 6th surface</td>
</tr>
<tr>
<td>s_6_4th</td>
<td>-2.57e-4</td>
<td>-1.41e-4</td>
<td>mm</td>
<td>4th order coefficient of polynomial fit to 6th surface</td>
</tr>
<tr>
<td>s_6_2nd</td>
<td>-4.99e-3</td>
<td>4.99e-3</td>
<td>mm</td>
<td>2nd order coefficient of polynomial fit to 6th surface</td>
</tr>
<tr>
<td>s_6_8th</td>
<td>-4.56e-8</td>
<td>4.92e-7</td>
<td>mm</td>
<td>8th order coefficient of polynomial fit to 6th surface</td>
</tr>
<tr>
<td>s_6_10th</td>
<td>-2.33e-8</td>
<td>8.80e-10</td>
<td>mm</td>
<td>10th order coefficient of polynomial fit to 6th surface</td>
</tr>
</tbody>
</table>

Table 2. Quantitative evaluation of end-to-end design and nominal design using simulated measurements on an unseen validation set for low-light imaging.

<table>
<thead>
<tr>
<th>Methods</th>
<th>1 - LPIPS</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end with Neural Network</td>
<td>0.961</td>
<td>35.6</td>
<td>0.942</td>
</tr>
<tr>
<td>Nominal with Neural Network</td>
<td>0.914</td>
<td>32.0</td>
<td>0.899</td>
</tr>
<tr>
<td>End-to-end with Hardware ISP</td>
<td>0.811</td>
<td>21.2</td>
<td>0.892</td>
</tr>
<tr>
<td>Nominal with Hardware ISP</td>
<td>0.750</td>
<td>21.1</td>
<td>0.871</td>
</tr>
</tbody>
</table>

end-to-end optimized optic features a similar support but different distribution without the elongated streak of the nominal design. We compare against the nominal optic using the expert-tuned settings for the hardware ISP, qualitative and quantitative results are shown in Fig. 8 and Tab. 2 respectively. Please see the supplemental document for further results.

**Single-image low-light imaging.** Low-light imaging is another important task which is affected by both the optics used for capture and the post-processing algorithms. For this experiment, we feed in RGB images from the MIT-Adobe FiveK dataset and scale down the exposure of the simulated Sony IMX249 by a 100x factor to 5μs, and then setting the gain factor in our sensor model $f_{\text{sensor}}$ to compensate for this scaling difference. We use the same compound optics, software ISP architecture, and loss as in the “image quality with software ISP” experiment. Note that both networks are able to learn to deconvolve the jointly optimized PSFs. Even for such fine-tuned processing, shown quantitatively in Tab. 3 and qualitatively in Fig. 9, our end-to-end pipeline demonstrates improved perceptual quality with substantially more fine detail preserved compared to fine-tuning the network $f_{\text{nn}}$ for the given the nominal optics. Please see the supplemental document for additional results.

**Automotive object detection and traffic light state detection with hardware ISP.** We now jointly optimize a full end-to-end pipeline for automotive object and traffic light detection, consisting of a Cooke triplet compound lens, ARM Mali-C71 ISP for image processing, and a Faster-RCNN [Ren et al. 2015] (with a ResNet-28 backbone) object detector, which we dub "FRCNN" for short in the following.

For object detection (OD), we rely on a training dataset created from BDD100K [Yu et al. 2020] by grouping different categories, resulting in 6 categories: car/van/suv, bus/truck/tram, person, bike, traffic lights, traffic signs. We use an additional 20000 images captured using a FLIR BFLY-23S6C-C camera with a Fujinon CF12.5HA lens to handle European scenes. For fair comparison against our end-to-end optimized pipeline, we fine-tuned the detector and ISP for the pipeline using the nominal optics on images simulated with the same nominal optics. We evaluate both pipelines on a validation set consisting of 10000 images from BDD100K and 10386 images from our additional captures. Quantitative metrics demonstrating improved object detection performance are shown in Tab. 4.

We also optimize the same pipeline for traffic light detection (TL). This time, we rely on the DriveU dataset [Fregin et al. 2018] and our own captures, again with fine-tuning for the nominal optics pipeline. For the labels, only 10 categories of front-facing traffic lights were considered: red circle/straight/left/right, green circle/straight/left/right, yellow circle, red-yellow circle. We evaluate both pipelines on a validation set consisting of 10905 images from DriveU and 1851 images from our additional captures. Quantitative metrics demonstrating improved traffic light detection performance are shown in Tab. 4. We refer to the supplemental document for qualitative results in simulation for both object detection and traffic light detection.

In both cases, the proposed method learns optics and processing pipelines that are different from the optics for perceptual image quality. These simulated optics follow the same trend as the manufactured prototype optics in Fig. 10, which we show here ahead of the detailed description in Sec. 7.1 for brevity. Specifically, both
the conventional (fine-tuned) ISP. The hardware ISP accentuates noise and shadow artifacts when attempting to compensate for the lower light efficiency of the nominal optics, which in turn results in reduced detection accuracy. In addition, the OD design favors a uniform PSF over all fields, while the TL lens exhibits a PSF with a slightly stronger peak component in the far periphery, aiding the detection of small details, e.g., traffic light arrows. Please see the supplemental document for further detail on the optimized optics and their design trade-offs compared to the nominal expert-designed optic.

### 6.2 Eight-element achromat experiments

We demonstrate that the applicability of our method extends to more complex compound lenses. Specifically, we repeat the "Image quality with hardware ISP" experiment for an 8-element achromat compound lens using the same optimization procedure as before. As this compound lens has many more degrees of freedom than the Cooke triplet, we expect to be able to learn nearly any spatial PSF suited towards our desired applications. The nominal lens design is well-corrected with small PSF spot sizes across all fields. Perhaps surprisingly, our experiments demonstrate that even for this complex lens system our approach retrieves compact PSFs that match and slightly improve upon the nominal PSFs, see qualitative results in Fig. 11. This is confirmed by the quantitative results in Tab. 5 which reports improvements in perceptual quality with the LPIPS an SSIM metric while keeping SNR the same. We refer to the supplemental document for additional results.

### 6.3 Validation of the optimization method

We demonstrate that our proposed optimization scheme described in Sec. 6.2 has superior optimization performance than vanilla end-to-end optimization for joint end-to-end optimization. In machine learning practice, the parameters of deep neural networks are often optimized using a vanilla stochastic gradient optimizer (e.g. SGD or Adam) which updates all trainable parameters with the same optimization settings (e.g. same learning rate) at each training iteration. While this is often sufficient for an isolated processing unit (with all others fixed), the proposed Alg. 1 is substantially less prone to local minima for our multi-stage imaging pipelines.

We perform a validation experiment by repeating the "Image quality with software ISP" and "Single-image low-light imaging" experiment from Sec. 6.1, but we now compare against the performance obtained when directly applying a vanilla stochastic gradient optimizer to all trainable parameters in a non-alternating fashion. For these comparison experiments we apply Alg. 1 by using the Adam optimizer with learning rate $10^{-2}$ for $P_{\text{optic}}$ and using the Adam optimizer with learning rate $10^{-4}$ for $P_{\text{nn}}$. Quantitative comparisons are shown in Fig. 12. We observed that using the same optimizer negatively impacted optimization performance. For the "Image quality with software ISP" experiment the vanilla optimizer became stuck in a local minima whereas our proposed optimization scheme successfully optimizes the imaging pipeline. For the "Single-image low-light imaging" experiment the vanilla optimizer
Differentiable Compound Optics and Processing Pipeline Optimization for End-to-end Camera Design

Object detection using Nominal Optics and ARM Mali-C71 ISP
Object detection using End-to-end Optimized Optics and ARM Mali-C71 ISP
Traffic light detection using Nominal Optics and ARM Mali-C71 ISP
Traffic light detection using End-to-end Optimized Optics and ARM Mali-C71 ISP

Fig. 10. Real-world prototype captures for automotive object and traffic light detection with Cooke triplet and hardware ISP. The manufactured prototypes are tested in the wild and demonstrate that our optimization allows for higher accuracy object and traffic light detection and classification. Note that our traffic light detector is trained to recognize vehicle traffic lights and ignores pedestrian traffic lights. The optimized optics have greater light efficiency (smaller f-number) and more uniform blur across the field of view than the nominal optic, which leads to greater detection performance. The optic optimized for traffic light detection is slightly sharper in the center and the peripheries than the optic optimized for object detection due to the size of traffic lights. Note that the object detection captures were taken during daytime whereas the traffic light detection captures were taken at dusk.

7 EXPERIMENTAL VALIDATION

7.1 Lens prototype manufacturing

We validate our proposed method by manufacturing five physical lens prototypes (two iterations for the nominal design, see Sec. 6.1, three obtained with our optimization procedure) and testing them on three different applications. With the manufacturing constraints and sensors available to us, we opted for a typical mid-to-far range automotive camera configuration using Cooke triplets with a field of view of $25^\circ$, allowing us to analyze image quality and detection of small objects at a distance typically affected the most by aberrations or ISP processing settings. We refer to the supplemental document for simulations with a larger field of view. All fabricated Cooke triplets have an effective focal length of 25 mm, and real clear aperture size of 5 mm (although optimized designs can stray from these initial values slightly). The designs comprise a negative flint glass element (Thorlabs LD2297, N-SF11 Bi-Concave Lens with AR coating) in the center with a positive polymethyl methacrylate (PMMA).

<table>
<thead>
<tr>
<th>Methods</th>
<th>1 - LPIPS</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-End with Hardware ISP</td>
<td>0.760</td>
<td>18.5</td>
<td>0.683</td>
</tr>
<tr>
<td>Nominal with Hardware ISP</td>
<td>0.728</td>
<td>18.5</td>
<td>0.675</td>
</tr>
</tbody>
</table>

Table 5. Quantitative evaluation of the eight-element lens designs from Fig. 11 using simulated measurements. As explained in the text, the margin of improvement is smaller than that of the Cooke triplet experiment because of the substantially greater degrees of freedom of the eight-element lens. Note that, nevertheless, the proposed optimization still manages to achieve slightly higher image quality. Note that the different field of views between the eight-element lens and the Cooke triplet results in different evaluation settings, thus these values are not comparable to those in Tab. 2.
element on each side. The two positive elements comprise an aspherical and a spherical surface, whose parameters are optimized by our approach. The substrate PMMA has a refractive index of 1.493 at the principle wavelength of 550 nm. This combination of materials with different Abbe numbers mitigates chromatic aberrations, while the use of aspherical surfaces provides more degrees of freedom to the optimization for achieving the desired optical behavior. Refer to the supplemental document for detailed specifications of all lenses.

To fabricate our customized lenses, we use a CNC machining system that supports 5-axis single point diamond turning (Nanotech 350FG) [Fang et al. 2013; Peng et al. 2019]. This process supports a high precision (≤ 1 µm) regarding the tolerance of the physical height of a continuous surface. We use standard mechanical turning to manufacture mounts, tubes, spacers, and physical barrels with aluminum alloy for assembling multiple optical elements, with a measured tolerance of 20 µm. In the design space, we empirically apply the constraints on the minimum air gap of 1 mm, the minimum edge thickness of optical elements of 2 mm, and the minimum back focal distance of 20 mm. A cross-section diagram of one representative lens is presented in the supplemental document. All lenses are assembled via C-mount to a FLIR BFLY-U3-23S6C-C camera with the same 2.3 megapixel Sony IMX249 sensor that was used for the synthetic experiments in the previous section (Sec. 6). To facilitate reproducibility we will provide all Zemax files and detailed lens manufacturing instructions.

7.2 Real-world validation of optimized Cooke triplets

We use the learned optics parameters $\mathcal{P}_c^{\text{opt}}$ obtained in simulation for three of the experiments shown in Sec. 6.1, namely image quality with hardware ISP, automotive object detection, and traffic light classification. After manufacturing each of the individual lens assemblies, we measure the spatially-varying PSF of the prototype lenses using a pinhole light source with a 75 µm pinhole diameter placed at 2 m from the camera. With the pinhole source at infinity focus, instead of moving the pinhole on a translation stage, we rotate the camera viewing angle while keeping the position fixed to measure the spatially-varying PSFs. We fine-tune all baseline and optimized downstream network and ISP blocks with these post-manufacturing
Table 7. Quantitative pedestrian-vehicle and traffic light detection evaluations using experimental measurements captured with the fabricated prototype lenses. We compare here nominal vs. our end-to-end optimized design, performed using the chart proposed in [Tseng et al. 2019]. In addition to PSNR and SSIM as conventional metrics, we also report 1 - LPIPS [Zhang et al. 2018] as a perceptual metric (higher is better).

<table>
<thead>
<tr>
<th>Methods</th>
<th>OD</th>
<th>TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end with Hardware ISP</td>
<td>0.598</td>
<td>17.25</td>
</tr>
<tr>
<td>Nominal with Hardware ISP</td>
<td>0.565</td>
<td>12.64</td>
</tr>
</tbody>
</table>

Table 6. Quantitative image quality evaluations using experimental measurements captured with the fabricated prototype lenses. Mean Average Precision (mAP) for object detection (OD) and traffic light (TL) state detection for our fabricated end-to-end optimized system versus the expert-designed nominal lens with detectors fine-tuned on captures from the same nominal optics.

<table>
<thead>
<tr>
<th>Methods</th>
<th>OD</th>
<th>TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end with Hardware ISP and FRCNN</td>
<td>46</td>
<td>31</td>
</tr>
<tr>
<td>Nominal with Hardware ISP and FRCNN</td>
<td>40</td>
<td>27</td>
</tr>
</tbody>
</table>

PSFs. Although identical RGB scenes can be used as input to different optical systems for the software experiments in Sec. 6, capturing identical frames with real lens systems is challenging—especially for the automotive experiments. For our testing setup, we placed two prototype lenses side-by-side; see the capture vehicle shown in Fig. 1. We did not use a beam-splitter setup as these can cause significant flare for HDR scenes. The two camera systems are synced using a hardware trigger and use the same fixed exposure for fair comparison. The results of each experiment are described next.

Image quality with hardware ISP. Similar to what was observed in simulation, our optimized compound lens reduces the aberrations that are present in the nominal compound lens. As such, the images acquired with our jointly optimized pipeline are superior to those acquired by separately tuning the optics and the hardware ISP. Qualitative results are shown in Fig. 13 and Fig. 14 and quantitative metrics are shown in Tab. 6.

Figs. 13 and 14 show a qualitative comparison of our optimized design against the nominal. When compared against the nominal compound lens, our end-to-end optimized optics and ISP demonstrate substantially sharper image quality in the peripheries and similar performance in the center, as evidenced by the color-checker and text inset in Fig. 13 and the city insets in Fig. 14. Tab. 6 validates this quantitatively, demonstrating that our optimized lens designs yield improved quantitative metrics (LPIPS, PSNR, and SSIM), computed on the custom chart from [Tseng et al. 2019].

Automotive object detection. For the fabricated optimized object detection lens and the nominal lens we performed synchronized dual-camera capture in a dense urban area in North America. We manually annotated 2005 dual camera pairs for evaluation with a total of 30,264 objects falling in the pedestrian and vehicle classes as described in Sec. 6. We use an unbiased team of annotators to separately annotate the nominal and target lens captures. Fig. 1 and Fig. 10 show example captures acquired and processed with the proposed system. In low-intensity regions, the captures and processed results with the nominal lens suffer from the lower light efficiency due the larger f-number (f/4.4) compared to the end-to-end learned optical system (f/3.2). The hardware ISP is not able to recover enough signal in these low-flux regions and instead amplifies measurement noise. As a result, even with slightly larger PSF, the proposed end-to-end learned system outperforms the nominal design in object detection. As a result of the spatial distribution and the object size, the learned optics prefer uniform aberrations across all field instead, which we attribute to the fact that it is detrimental for the convolutional detector to learn spatially-varying processing. Fig. 1 and Fig. 10 validate these characteristics and shows examples where the nominal system fails to detect low-light edge boundaries between objects (e.g., parked row of cars are often missed). Object detection with the nominal lens misses pedestrians and small objects with complex background (Fig. 1 center left). We note that the object detection network architecture for our end-to-end pipeline is the same as the simulation one from Sec. 6 except with the optics and the sensor simulation layers removed for inference on captured data. The results in Tab. 7 validate that the proposed system performs significantly better in terms of 2D mean average precision.

Traffic light state detection. Using the same synchronized dual-camera setup we validate the proposed approach using our end-to-end optimized traffic light state detection lens compared to the nominal (fine-tuned) system. For the assessment, we annotated 2264 dual-camera captures with a total of 8442 traffic lights with states annotated using the same labels as described in Sec. 6. Similar to the OD lens design, the TL lens is faster (f/3.3) than the nominal design (f/4.4), resulting in substantially improved SNR, which improves detections especially in low-flux regions where the (fine-tuned) nominal ISP is not capable of recovering enough signal. As a result of the spatial distribution and size of the small traffic lights, the traffic light lens differs from the previous lens for pedestrian-vehicle detection. Specifically, the TL lens exhibits a PSF with small spot-size in the center, where small traffic lights at a distance appear, and it has a PSF with a peak component in the periphery, where closer traffic lights appear in the upper periphery of the sensor. Compared to the nominal design, this peak component results in sharper details in the periphery. Fig. 10 shows examples where improved sharpness and the lower f-number aid the detection of small traffic lights especially in challenging scenes with low ambient illumination. Here, for the nominal lens the arrows appear circular and are often too blurred to be detected. Tab. 7 validates that our end-to-end optimized optics and processing outperforms the nominal system also quantitatively on the captured validation set.

8 DISCUSSION AND CONCLUSION

Limitations. Our method does not replace human optics and software engineers in the end-to-end design process—analogously, neither do ZEMAX or TENSORFLOW/PyTorch for optical design or machine learning design. Rather, our approach augments compartmentalized camera design tools by bridging a longstanding gap between heterogeneous sensing, compute, and algorithm design. Furthermore, end-to-end optimization is fundamentally limited by the availability of real-world data needed to simulate image formation and end-to-end task losses, e.g., an IoU detection loss. Given that larger
training corpora of realistic high-resolution multi-spectral data are not readily available, we concentrate on RGB optical designs in the optical forward model. We also assumed the scene to be at infinity, which precludes applicability on depth-sensitive applications such as mobile photography for which auto-focus is necessary.

**Conclusion.** This paper introduces a framework for joint end-to-end optimization of a compound lens model together with a realistic sensor model, hardware (or software) ISP, and downstream CNN computer vision module. We jointly train all parameters and hyperparameters of this heterogeneous camera pipeline for a domain-specific loss. The proposed framework builds on traditional tolerance analysis and seamlessly integrates with traditional optics design methods. Based on the optimized optical parameters obtained from our fully-differentiable imaging pipeline, we build five prototype compound lens designs and assess them on real-world driving data for automotive camera design. We validate the proposed method on alternative optics and post-processing for human viewing in challenging outdoor and low-light scenarios. We also validate the method for automotive camera optics together with hardware ISP post-processing and detection, beating state-of-the-art self-driving vehicle camera designs. In all applications, the approach outperforms existing compartmentalized design or fine-tuning qualitatively and quantitatively on all domain-specific applications tested. Furthermore, we have demonstrated that producing high-quality images for human viewing is not a necessary or even desirable constraint for machine vision applications such as automotive object detection.

Possible future directions include incorporating automatic neural architecture search [Elsken et al. 2019] and lens design search to circumvent the requirement that the number of lens elements must be specified a priori. Finally, automating sensor design, active illumination, and fusion with different sensors, e.g. in multi-camera acquisition systems, are exciting avenues for future research that this work makes first step towards.
We thank our reviewers for their invaluable comments. We thank Adam Finkelstein, Szymon Rusinkiewicz, and Hatem Ben Zakour for helpful discussions. Felix Heide was supported by an NSF CAREER Award (2047359) and a Sony Faculty Innovation Award.

REFERENCES


ACM Transactions on Graphics, Vol. 40, No. 0, Article 0. Publication date: August 2021.