

Towards self-calibrated lens metrology by differentiable refractive deflectometry

CONGLI WANG, NI CHEN, AND WOLFGANG HEIDRICH*

Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia

*wolfgang.heidrich@kaust.edu.sa

Abstract: Deflectometry, as a non-contact, fully optical metrology method, is difficult to apply to refractive elements due to multi-surface entanglement and precise pose alignment. Here, we present a computational self-calibration approach to measure parametric lenses using dual-camera refractive deflectometry, achieved by an accurate, differentiable, and efficient ray tracing framework for modeling the metrology setup, based on which damped least squares is utilized to estimate unknown lens shape and pose parameters. We successfully demonstrate both synthetic and experimental results on singlet lens surface curvature and asphere-freeform metrology in a transmissive setting.

© 2021 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Refractive lens metrology plays an important role in lens manufacturing, quality control, and reverse engineering. Various techniques have been proposed for general-purpose freeform optical surface metrology. As a non-interferometric method, deflectometry maintains a strong interest [1] for its simplicity, low-cost, and insensitivity for environments, from specular surface measurement using computerized phase shifting screens [2–4] to 2D/3D tomography refractive index reconstruction using background oriented Schlieren [5–8] and variants [9, 10]. However, it is difficult to directly apply deflectometry to refractive elements, for the following reasons: (i) Multi-surface interactions mean that deflection shifts are ambiguous and cannot be attributed to a single surface; (ii) Lens alignment is sensitive and misalignment may dominate the actual surface deviations from nominal setup; (iii) Basis representation is required for freeform surfaces; (iv) Strong refraction leads to large deflections for large-curvature surfaces. There are works demonstrated for multi-surface reconstruction [11, 12], as well as efforts in iterative optimization [13] and with self-calibration [14–17], pose estimation [18], model-based fitting [19], dual-camera setups [20, 21], adaptive null testing [22], and lens power metrology using refractive deflectometry [23]. However, these techniques are aimed at different applications, and cannot fully address all the issues in refractive deflectometry.

Given the metrology measurements (e.g., phase-shifted images, ray intersections, transmitted wavefront maps), the task is to estimate a set of unknown parameters characterizing the lens surfaces that fit the measurements reasonably well. This problem of refractive lens testing can be generally rephrased as an inverse problem. Thus, an automated, transparent, controllable data analysis process is desired, as a general-purpose computational solution. Compared to the rapid development of new instruments, computational techniques remain relatively unexplored, we believe current metrology techniques could be further improved by advanced computational methods. Of these methods, gradient-based algorithms are of particular interest for finding a solution to the problem. However, gradient evaluation is sometimes infeasible using a finite-difference approximation which scales up with the number of variables. Analytic derivations may not oftentimes be desirable because of the potentially lengthy formulas [24]. Automatic differentiation, in which gradients are computed automatically, has been proposed for solving problems efficiently in phase retrieval [25], lens design [26–30],

phase microscopy [31], ptychography [32, 33], head-mounted displays calibration [34], and transparent object reconstruction [35]. By forward modeling the measurements using an automatic differentiation engine, a computation graph is constructed, to which the gradients can be evaluated up to numerical precision by repeatedly applying the chain rule [25, 36]. Given the gradients, starting from a proper initial guess, unknown model parameters can be iteratively updated. This approach is a general solution to inverse problems, including our problem of interest in this paper.

Direct application of automatic differentiation to our metrology problem is not practical using existing frameworks such as PyTorch [37], because of a few missing blocks: (i) Accuracy. Metrology requires accurate modeling of the measurement setup by ray tracing, from camera to the target lens, and to display screen, with correct shape and pose parameterization. (ii) Differentiability. The ray tracer needs to be differentiable, such that gradients can be evaluated by back-propagating the acquired data to the unknown lens parameters. (iii) Efficiency. A memory- and computationally efficient pipeline satisfying (i)(ii), for a specially crafted inverse solver to take advantage of.

In this paper, we introduce differentiable refractive deflectometry, a new technique based on automatic differentiation. To mitigate phase ambiguity, we employ a dual-camera refractive deflectometry hardware setup, where screen intersections can be obtained by phase-shifting the display screen (Section 2.1). We build and model the physical metrology setup described by shape and pose parameters, by ray-tracing multiple refractive surfaces using Snell’s law. Differentiability is provided using automatic differentiation, yet to ensure efficiency we propose a differentiable root finder to compute ray-surface intersection (Section 2.2). Utilizing the gradient information, inverse estimation and analysis can be performed, where damped least squares [38] is employed to solve for lens parameters from measurement intersections in a self-calibrated way (Section 2.3), and an uncertainty analysis provides a criterion for understanding solution stability (Section 2.4). Both synthetic (Section 3.1) and experimental (Section 3.2) results validate our approach. We believe the proposed framework provides a new, general, extensible, and reproducible computational solution to automated data analysis for existing and future deflectometry techniques. Source code and examples will be available at [39].

2. Method

2.1. Problem formulation

The metrology process involves both hardware image acquisition and software, as in Figure 1. The setup is based on the phase measuring deflectometry [2], but reconfigured in a refractive mode using a dual-camera setup to introduce view-variant phase measurements for mitigation of surface ambiguity, because both refractive surfaces contribute to the measurable phase. Figure 1(a) shows the schematic diagram. A programmable display screen (Apple MacBook Pro 13.3”, pixel pitch 111.8 μm) shows a set of 90° phase-shifted sinusoidal patterns, with the target lens placed in front of the screen. Two cameras of F-number $f/16$ (FLIR GS3-U3-50S5M, pixel pitch 3.45 μm) are employed to take grayscale images as in Figure 1(b). Gamma-correction is applied to ensure a linear relationship between screen and image pixel values. Grayscale images of different phase-shifts ($I_{0^\circ}, I_{90^\circ}, I_{180^\circ}, I_{270^\circ}$) are obtained, using 90° phase-shifts. These images encode refractive directional information regarding the testing lens, and hence are processed using the standard four-step phase shifting method followed by phase unwrapping [40], to retrieve observed screen locations $\hat{\mathbf{p}} = (p_x, p_y)$:

$$p_{x,y} = \text{unwrap} \left(\arctan \left(\frac{I_{270^\circ} - I_{90^\circ}}{I_{0^\circ} - I_{180^\circ}} \right) \right). \quad (1)$$

This fringe analysis process repeats for x and y respectively to obtain p_x and p_y , and repeats again separately for both cameras $i = 1, 2$, obtaining intersection points $\hat{\mathbf{p}}_1$ and $\hat{\mathbf{p}}_2$, as inputs to next step data analysis in Figure 1(d).

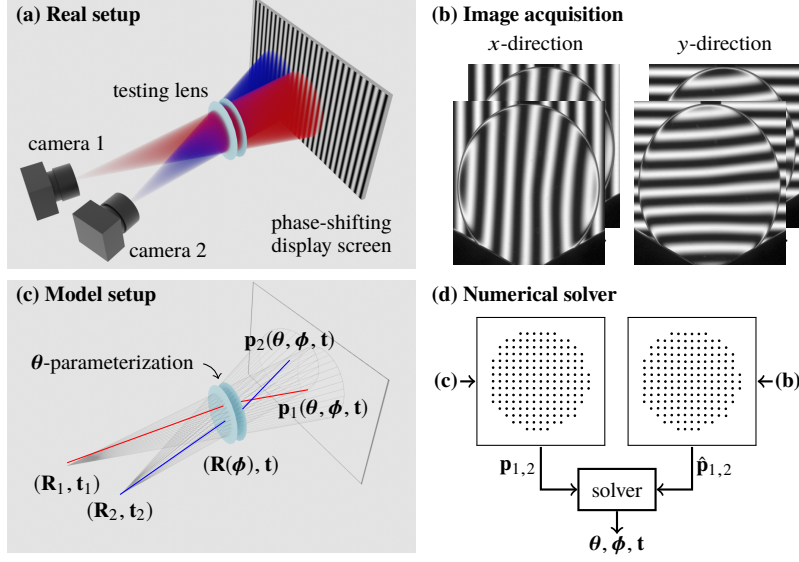


Fig. 1. Dual-camera refractive deflectometry for lens metrology. **(a)** Hardware setup. **(b)** Captured phase-shifted images, from which on-screen intersections $\hat{\mathbf{p}}_i$ ($i = 1, 2$) are obtained. **(c)** A ray tracer models the setup by ray tracing each parameterized refractive surface, obtaining the modeled intersections $\mathbf{p}_i(\theta, \phi, \mathbf{t})$. **(d)** Unknown θ and pose (ϕ, \mathbf{t}) are jointly optimized by minimizing the error between \mathbf{p}_i and $\hat{\mathbf{p}}_i$.

The target lens is assumed to be parameterized by θ , with an unknown pose (ϕ, \mathbf{t}) . Possible θ parameterizations are lens curvatures, or freeform coefficients, see Appendix A for details. A ray tracer models the setup as in Figure 1(c), resulting in modeled intersections $\mathbf{p}_i(\theta, \phi, \mathbf{t})$. Given \mathbf{p}_i and $\hat{\mathbf{p}}_i$, a numerical solver in Figure 1(d) jointly estimates θ, ϕ, \mathbf{t} such that an error metric (also known as the loss function) is minimized, yielding the metrology values θ^* . Here, we minimize the least squared error:

$$\theta^*, \phi^*, \mathbf{t}^* = \arg \min_{\theta, \phi, \mathbf{t}} \sum_{i=1}^2 \|\mathbf{p}_i(\theta, \phi, \mathbf{t}) - \hat{\mathbf{p}}_i\|_2^2. \quad (2)$$

We optimize Eq. (2) using damped least squares [38], as will be detailed in Section 2.3. The following elaborates on computation for $\mathbf{p}_i(\cdot)$, the modeling.

2.2. Modeling metrology setup

Each object (camera, target lens, screen) is associated with a rigid transformation (\mathbf{R}, \mathbf{t}) in world coordinate (screen's frame), with a rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. Ray tracing is performed in local frames. For pose estimation, the target lens transformation $(\mathbf{R}(\phi), \mathbf{t})$ can be determined from the six degree-of-freedom parameters (ϕ, \mathbf{t}) where $\phi \in \mathbb{R}^3$ are the rotation angles around (x, y, z) axes. Cameras are treated as perspective pinholes, with the intrinsic and extrinsic parameters obtained from calibration [41].

Rays are generated for each image pixel by the following procedure:

1. Sampling from camera pinhole model;
2. Intersecting surfaces of the target lens via a root finder, refraction and deflection by Snell's law, and hence the outgoing rays are associated with lens parameters θ and pose $(\mathbf{R}(\phi), \mathbf{t})$;

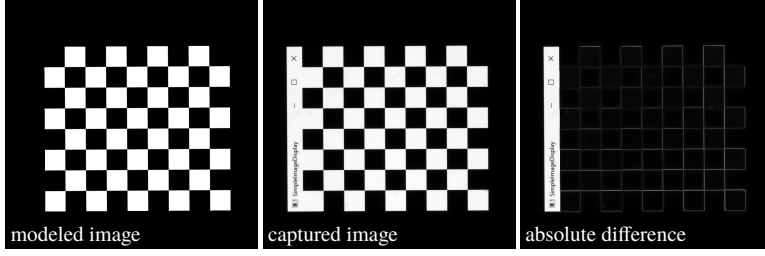


Fig. 2. Assuming pinhole camera model and planar screen, our ray tracer reproduces captured images in high fidelity.

3. Reaching towards a presumably planar display screen, obtaining the modeled intersections $\mathbf{p}_i(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{t})$ for both cameras.

Of these steps, a root finder to determine the ray-surface intersection is the most critical component for aspheres and freeform surfaces, which do not have closed-form analytical solutions. The intersection problem requires solving for intersection (x, y, z) and a ray marching distance t for optical surface $f(x, y, z; \boldsymbol{\theta}) = 0$, given a ray (\mathbf{o}, \mathbf{d}) of origin $\mathbf{o} = (o_x, o_y, o_z)$ and direction $\mathbf{d} = (d_x, d_y, d_z)$ of unit length. Mathematically:

$$\text{find } t > 0 \quad \text{subject to } f(x, y, z; \boldsymbol{\theta}) = f(\mathbf{o} + t\mathbf{d}; \boldsymbol{\theta}) = 0. \quad (3)$$

Equation (3) can be solved using an iterative root finder, with the iterations directly unrolled for gradients, so that ray marching distances t can be related through lens parameters $\boldsymbol{\theta}$ via automatic differentiation. However, this straightforward approach is not efficient and is memory consuming because of storing the intermediate iteration variables and their derivatives for every ray in the image. Fortunately, there is an analytic approach for the desired gradient to be computed *outside* of automatic differentiation. Denote the solution to Eq. (3) as t^* , and exploit the implicit function theorem for differentiation w.r.t. $\boldsymbol{\theta}$:

$$f(\mathbf{o} + t^*(\boldsymbol{\theta})\mathbf{d}; \boldsymbol{\theta}) = 0 \quad \text{and} \quad (\nabla f \cdot \mathbf{d}) \frac{\partial t^*}{\partial \boldsymbol{\theta}} + \frac{\partial f}{\partial \boldsymbol{\theta}} = 0. \quad (4)$$

Rearranging above, yields an analytic formula for gradient:

$$\frac{\partial t^*}{\partial \boldsymbol{\theta}} = -\frac{1}{\nabla f \cdot \mathbf{d}} \frac{\partial f}{\partial \boldsymbol{\theta}}. \quad (5)$$

In other words, we can first compute t^* without automatic differentiation (and no intermediate variables stored), and then amend its gradient back to automatic differentiation by Eq. (5). We employ Newton's method for obtaining t^* , initialized by a non-singular estimate $t^{(0)} = (z_f - o_z)/d_z$, with iteration stops when the residual is smaller than tolerance. The method converges in a few iterations. When properly setup, with no target lens present, the modeled image matches the real image reasonably well as in Figure 2, demonstrating the physical realism of the ray tracer. Blurry edges are from diffraction, which is ignored in this paper, since we describe a ray optics approach.

2.3. Optimization using damped least squares

In this sub-section, we slightly abuse the notation by letting $\boldsymbol{\theta} := (\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{t})$. Our target is to estimate $\boldsymbol{\theta} \in \mathbb{R}^n$ in Eq. (2) for the following using damped least squares [38]:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^2 \|\mathbf{p}_i(\boldsymbol{\theta}) - \hat{\mathbf{p}}_i\|_2^2, \quad (6)$$

where recall that function $\mathbf{p}_i(\cdot): \mathbb{R}^n \mapsto \mathbb{R}^m$ maps from the n unknown parameters to the m modeled intersection points, and $m \gg n$. To obtain an iterative solution to Eq. (6), at current estimate $\boldsymbol{\theta}^{(k)}$, we solve for a small change of $\Delta\boldsymbol{\theta}$ to obtain $\boldsymbol{\theta}^{(k+1)}$, such that:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}, \quad \text{where } \Delta\boldsymbol{\theta} = \arg \min_{\Delta\boldsymbol{\theta}} \sum_{i=1}^2 \|\mathbf{p}_i(\boldsymbol{\theta}^{(k+1)}) - \hat{\mathbf{p}}_i\|_2^2. \quad (7)$$

The iteration stops when a stopping criterion is met. At each step k , each non-linear function $\mathbf{p}_i(\cdot)$ is approximated using a first-order Taylor expansion:

$$\mathbf{p}_i(\boldsymbol{\theta}^{(k+1)}) = \mathbf{p}_i(\boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}) \approx \mathbf{p}_i(\boldsymbol{\theta}^{(k)}) + \mathbf{J}_i \Delta\boldsymbol{\theta}, \quad \text{where Jacobian } \mathbf{J}_i = \frac{\partial \mathbf{p}_i}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{m \times n}. \quad (8)$$

With Eq. (8), the least squares problem in Eq. (7) can be solved using the damped normal equation to ensure robustness, by adding a diagonal matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ scaled by an iteratively changing damping factor $\rho > 0$:

$$\left(\sum_{i=1}^2 \mathbf{J}_i^T \mathbf{J}_i + \rho \mathbf{D} \right) \Delta\boldsymbol{\theta} = \sum_{i=1}^2 \mathbf{J}_i^T (\hat{\mathbf{p}}_i - \mathbf{p}_i(\boldsymbol{\theta}^{(k)})). \quad (9)$$

In practice, we found \mathbf{D} being an identity matrix yields satisfactory results.

Features of automatic differentiation can be employed to efficiently solve the linear system in Eq. (9). Notice that the Jacobian \mathbf{J}_i is a tall matrix (since $m \gg n$), and it can be efficiently evaluated in a column-wise fashion using the forward-mode, hence obtaining $\mathbf{J}_i^T \mathbf{J}_i$ on the left-hand-side, whereas the right-hand-side can be obtained by reverse-mode without explicitly constructing \mathbf{J}_i^T . These speed improvements in terms of efficiency would not be possible without automatic differentiation.

Our framework and the solver are implemented in PyTorch [37], as the automatic differentiation framework provides a straightforward way to accurately evaluate gradients. For a small set of $\boldsymbol{\theta} \in \mathbb{R}^n$ when $n < 20$, the solver converges in a few seconds on a modern GPU, indicating practical usage of the method for fast metrology applications. Further speed improvements are possible using optimized high-performance code or on advanced computing platforms.

2.4. Uncertainty variance analysis

We can perform an uncertainty analysis on Eq. (6) to understand solution stability. Analyzing the derivatives (ignoring constants):

$$\partial \text{loss} = \sum_{i=1}^2 (\mathbf{p}_i(\boldsymbol{\theta}) - \hat{\mathbf{p}}_i) \cdot \mathbf{J}_i \partial \boldsymbol{\theta} = \mathbf{v} \cdot \partial \boldsymbol{\theta}, \quad \text{where } \mathbf{v} = \sum_{i=1}^2 \mathbf{J}_i^T (\mathbf{p}_i(\boldsymbol{\theta}) - \hat{\mathbf{p}}_i). \quad (10)$$

With an independence assumption on $\boldsymbol{\theta} \in \mathbb{R}^n$ (when properly parameterized) and equal prior probabilities, denoting $\text{diag}(\cdot)$ as a diagonal matrix formed by the corresponding vector, the uncertainty variance of each element of variable $\boldsymbol{\theta}$, is hence calculated as:

$$\sigma_{\text{loss}}^2 = \mathbf{v}^T \text{diag}(\sigma_{\boldsymbol{\theta}}^2) \mathbf{v} \quad \Rightarrow \quad \sigma_{\boldsymbol{\theta}}^2 = \frac{\sigma_{\text{loss}}^2}{n|\mathbf{v}|^2}, \quad (11)$$

where σ_{loss}^2 can be calculated from the optimal point, and $|\mathbf{v}|^2$ is the element-wise squared of the vector \mathbf{v} .

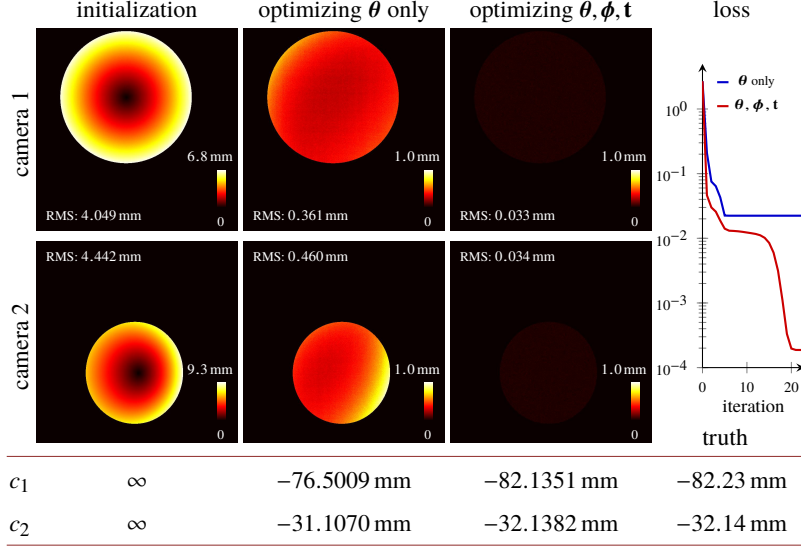


Fig. 3. Lens curvature $\theta = (c_1, c_2)$ metrology using synthetic data. Intersection error maps $|\mathbf{p}_i - \hat{\mathbf{p}}_i|$ are shown. Metrology data analysis is sensitive to minor misalignment, and a self-calibration approach is preferable by jointly optimizing θ, ϕ, \mathbf{t} .

3. Results

3.1. Synthetic results

Verification experiments are performed in simulation to verify the proposed self-calibration method. Figure 3 simulates a convex-concave lens (Thorlabs LE1234) under metrology test for curvature estimation, where the desired parameters $\theta = (c_1, c_2)$ are the two surface curvatures c_1 and c_2 . The lens suffered from a minor misalignment perturbation $\phi = (-0.3^\circ, 0.5^\circ, 0^\circ)$ and $\mathbf{t} = \mathbf{t}_0 + \Delta\mathbf{t}$ where \mathbf{t}_0 is obtained from triangulation but \mathbf{t} is deviated by $\Delta\mathbf{t} = (0.5 \text{ mm}, 0.5 \text{ mm}, 0.5 \text{ mm})$. Synthetic images were corrupted by Gaussian noise. When assuming perfect alignment ($\phi = \mathbf{0}$, $\Delta\mathbf{t} = \mathbf{0}$, i.e., only optimizing θ), the solver cannot be employed to predict the correct curvatures because of misalignment systematic errors in the measurement, whereas a self-calibration estimation (ϕ, \mathbf{t}, θ) is successful in error reduction, and a more accurate curvature estimate as in the table. This is illustrated in the final errors of the two methods, that the self-calibration approach (optimizing θ, ϕ, \mathbf{t}) produces a final error of more than a magnitude smaller compared to the non self-calibrated approach optimizing θ only. This demonstrates the superiority of joint shape and pose parameter estimation, which would not be viable without our accurate modeling of the setup.

Similar conclusion also holds for freeform metrology, as in Figure 4 where an asphere-freeform lens is simulated, suffered from the same misalignment (ϕ, \mathbf{t}), where the task is to estimate the cubic B-spline freeform surface coefficients θ , knowing the asphere profile. This one-surface constraint is necessary to limit the solution space, in that two arbitrary freeform surfaces can easily over-fit $\hat{\mathbf{p}}_i$, while being rarely practical in reality. When optimizing only θ , the misalignment affects metrology results. In contrast, a joint optimization of both θ and (ϕ, \mathbf{t}) significantly reduces the error, showing the benefit of a self-calibration approach.

3.2. Experimental results

Experimental results are supportive. Figure 5 shows experimental results on lens curvature metrology for a convex-concave lens (Thorlabs LE1234). The lens was mounted and placed in

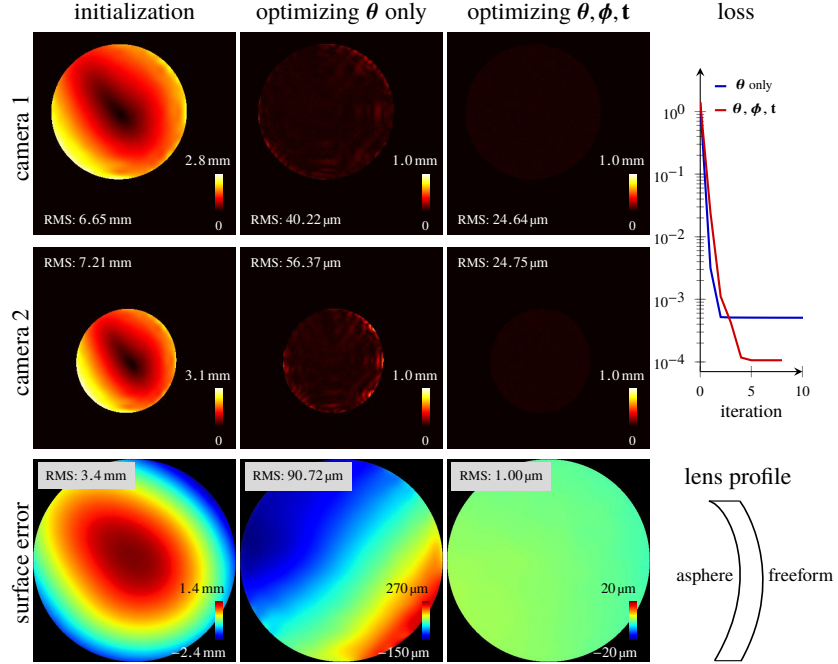


Fig. 4. Freeform lens metrology using synthetic data. Intersection error maps $|\mathbf{p}_i - \hat{\mathbf{p}}_i|$ and freeform surface error are shown. A self-calibration approach produces better results.

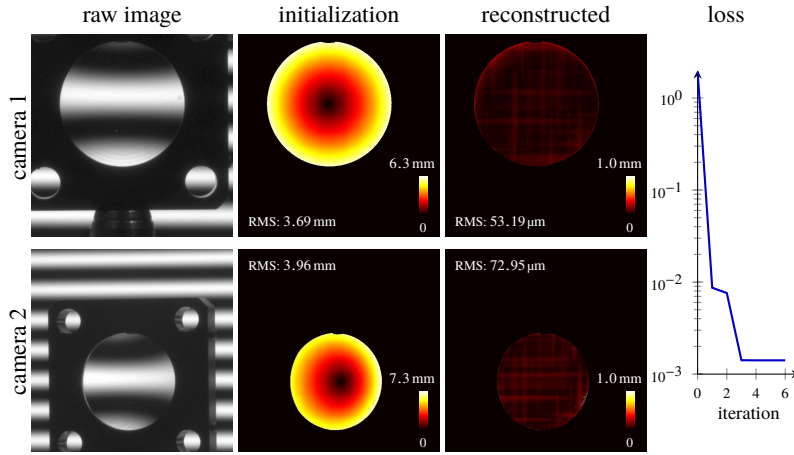


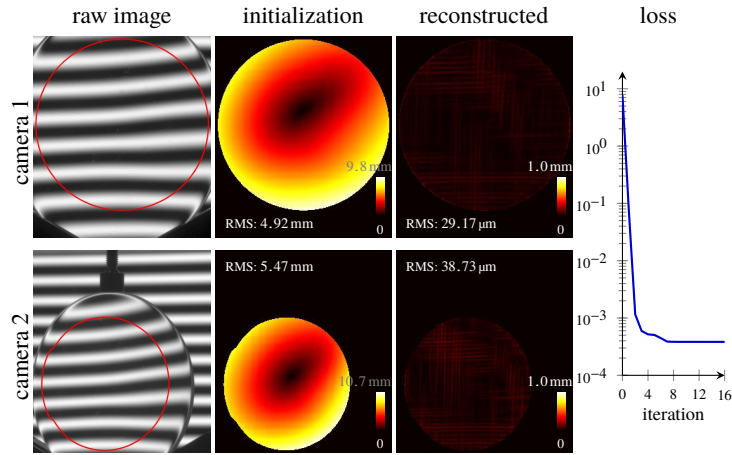
Fig. 5. Experimental lens curvature metrology for LE1234.

front of the screen at a distance of approximately 5 cm. The target lens pose $(\phi, \mathbf{t}_0 + \Delta\mathbf{t})$ is assumed to be in perfect angular alignment $\phi = \mathbf{0}$, but suffered from minor displacement errors ($\Delta\mathbf{t} \neq \mathbf{0}$), with the nominal origin \mathbf{t}_0 computed from dual-camera triangulation. Initialized from planar surfaces ($c_1 = c_2 = \infty$), our optimized curvatures are close to the manufacturer design parameters (Table 1), though the fitting error increases slightly at lens boundary. Two other Thorlabs lenses were under the same test, but with both ϕ and $\Delta\mathbf{t}$ optimized. Surface curvatures metrology results are shown in Table 1, our metrology results were close to the nominal manufacturer's

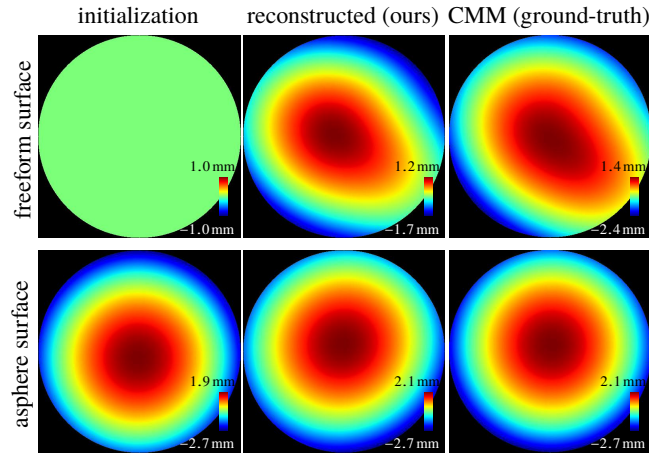
specifications, demonstrating the feasibility of our method to measure multi-surface curvatures.

Table 1. Experimental singlet lens curvature metrology results.

[mm] / lens name	LA1986	LE1234	LF1822
focal length	125	100	-100
ours (c_1, c_2)	(62.10, 828.39)	(-82.41, -32.61)	(34.68, 101.41)
truth (c_1, c_2)	(64.38, ∞)	(-82.23, -32.14)	(33.65, 100.00)



(a) Raw images and intersection error maps



(b) Reconstruction surfaces

Fig. 6. Experimental asphere-freeform lens metrology. (a) Raw images with the regions of interest (contoured by red lines) and optimization intersection errors. (b) Reconstruction comparison against CMM metrology results as ground truth.

Freeform lens experimental metrology results are also encouraging. The target optics is an asphere-freeform lens, whose two surfaces were discretely sampled and measured by a coordinate measuring machine (CMM) machine as ground truth. This freeform lens was mounted and placed

approximately 5 cm in front of the screen for measurement (see Figure 1(b)). As in simulation, we assume that an approximate surface profile of the asphere is given, and would like to solve for the freeform surface, with the pose unknown. Results are shown in Figure 6. Our solver optimizes the metrology data to a small residual intersection errors for both cameras, as in Figure 6(a). Though wrongly initialized for the asphere surface, the solver optimizes back to the correct orientation, and the reconstruction surfaces are visually similar to the ground truth in Figure 6(b), but are spatially transformed due to different alignment in deflectometry experiment and CMM data metrology, and double-surface entanglement as well in that the solution space is too huge to over-fit the data. Yet, our method provides an initial method to qualitatively profile both surfaces, when there is limited prior knowledge available. We have demonstrated the capability of the proposed method for surface metrology, especially for surface curvature estimations.

4. Discussion and Conclusion

Given current results, several avenues for future improvements are apparent. The underlining principle of this paper shares a high similarity with methods in profilometry, from which common issues such as influence of phase coding and variation of light transmittance can be analyzed and resolved [42]. On the hardware side, current image acquisition pipeline could be extended to a multi-angle tomography setup, or encode/decode intersections instantaneously to improve acquisition speed [43]. Doublets are also possible by incorporating ours into a data fusion pipeline [44]. In software, thanks to automatic differentiation, attainable gradient information allows for a family of solvers to be employed for accelerated convergence, compromising different trade-off factors. We may also extend ray-surface interaction beyond pure refraction and take optical aberrations into consideration [45, 46]. Suitable parameterization is also important for a full characterization of optical elements. Lastly, uncertainty analysis deserves more attention, as deflectometry itself requires a computationally heavy procedure which may introduce data misinterpretation.

In conclusion, we have demonstrated differentiable refractive deflectometry for self-calibrated lens metrology. Given the phase-shifting images, a fringe analysis provides measurement intersection points for the method to proceed, where both the unknown lens parameters and the pose are jointly optimized using a differentiable ray tracer. We believe the opened up new computational possibilities for lens metrology data analysis and other relevant application areas.

Funding. King Abdullah University of Science and Technology (Individual Baseline Funding, Center Partnership Funding).

Acknowledgments. The authors thank Carsten Glasenapp and Ivo Ihrke from Carl Zeiss AG for providing the freeform lens and its surface data.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available in Ref. [39].

A. Lens parameterization

A.1. Surface parameterization

We parameterize each lens surface in the implicit form:

$$f(x, y, z; \theta) = 0. \quad (12)$$

We consider three specific types of parameterized lens surfaces to represent lens and freeform surfaces, yet in theory alternative parameterization forms (see [47]) should also work. Surfaces are defined in a Cartesian coordinate system (x, y, z) , with z -axis being chosen as the optical axis (if any).

A.1.1. Aspheres

Let $\rho = x^2 + y^2$ since aspheric surfaces are axially symmetric. The sag distance function $s(\rho)$ of aspheric surfaces and its derivative with respect to ρ are:

$$s(\rho) = \frac{c\rho}{1 + \sqrt{1 - \alpha\rho}} + \sum_{i=2}^n a_{2i}\rho^i, \quad (13)$$

$$s'(\rho) = c \frac{1 + \sqrt{1 - \alpha\rho} - \alpha\rho/2}{\sqrt{1 - \alpha\rho} (1 + \sqrt{1 - \alpha\rho})^2} + \sum_{i=2}^n a_{2i}i\rho^{i-1}, \quad (14)$$

where c is the curvature, $\alpha = (1 + \kappa)c^2$ with κ being the conic coefficient, and a_{2i} 's are higher-order coefficients. Spherical surfaces are special cases of aspheric surfaces, with $\kappa = 0$ and $a_{2i} = 0$ ($i = 2, \dots, n$). In implicit form:

$$f(x, y, z; \boldsymbol{\theta}) = s(\rho) - z, \quad (15)$$

$$\nabla f(x, y, z; \boldsymbol{\theta}) = (2s'(\rho)x, 2s'(\rho)y, -1), \quad (16)$$

where differentiable parameters $\boldsymbol{\theta} = (c, \kappa, a_{2i})$.

A.1.2. XY polynomials

XY polynomial surfaces extend lens surface representation beyond axial symmetry. The implicit surface function $f(x, y, z; \boldsymbol{\theta})$ and its spatial derivatives are:

$$f(x, y, z; \boldsymbol{\theta}) = \sum_{j=0}^J \sum_{i=0}^j a_{i,j} x^i y^{j-i} + bz^2 - z, \quad (17)$$

$$\nabla f(x, y, z; \boldsymbol{\theta}) = \left(\sum_{j=1}^J \sum_{i=0}^j a_{i,j} i x^{i-1} y^{j-i}, \sum_{j=1}^J \sum_{i=0}^j a_{i,j} (j-i) x^i y^{j-i-1}, 2bz - 1 \right), \quad (18)$$

where differentiable parameters $\boldsymbol{\theta} = (b, a_{i,j})$.

A.1.3. B-splines

We employ B-splines [48] to represent high degree-of-freedom freeform surfaces. In general, the sag distance function $g(x, y)$ is represented as a spline of degree (in our case, it is three, i.e. the cubic B-spline) on the rectangle area, with predefined number of knots and knot positions. With that, spline functions $S_{i,j}(x, y)$ are fixed, and $g(x, y)$ is determined by spline coefficients $c_{i,j}$:

$$f(x, y, z; \boldsymbol{\theta}) = \sum_i^n \sum_j^m c_{i,j} S_{i,j}(x, y) - z, \quad (19)$$

$$\nabla f(x, y, z; \boldsymbol{\theta}) = \left(\sum_i^n \sum_j^m c_{i,j} \nabla_x S_{i,j}(x, y), \sum_i^n \sum_j^m c_{i,j} \nabla_y S_{i,j}(x, y), -1 \right), \quad (20)$$

where differentiable parameters $\boldsymbol{\theta} = (c_{i,j})$, and the spatial gradients of the spline functions $\nabla_x S_{i,j}$ and $\nabla_y S_{i,j}$ are efficiently evaluated via modified de-Boor's algorithm [48].

A.2. Lens type

In simulation and experiments, two types of lenses were considered:

- Singlet spherical lens, where the two surface curvatures are of interest, and $\theta = (c_1, c_2)$.
- Asphere-freeform lens, where one surface is asphere (parameterized by XY polynomial coefficients θ_{XY}), while another is cubic B-spline freeform surface (parameterized by θ_{spline}). In simulation $\theta = \theta_{\text{spline}}$ and in experiment $\theta = (\theta_{XY}, \theta_{\text{spline}})$.

We also assumed the following parameters are known: (i) lens diameter (ii) center thickness (iii) material refractive index at 562 nm. Although these additional parameters could also be jointly optimized using the proposed framework.

References

1. C. Faber, E. Olesch, R. Krobot, and G. Häusler, "Deflectometry challenges interferometry: The competition gets tougher!" in *Interferometry XVI: Techniques and Analysis*, vol. 8493 (International Society for Optics and Photonics, 2012), p. 84930R.
2. M. C. Knauer, J. Kaminski, and G. Hausler, "Phase measuring deflectometry: A new approach to measure specular free-form surfaces," in *Optical Metrology in Production Engineering*, vol. 5457 (International Society for Optics and Photonics, 2004), pp. 366–376.
3. P. Su, R. E. Parks, L. Wang, R. P. Angel, and J. H. Burge, "Software configurable optical test system: A computerized reverse hartmann test," *Appl. Opt.* **49**, 4404–4412 (2010).
4. F. Willomitzer, C.-K. Yeh, V. Gupta, W. Spies, F. Schiffrers, A. Katsaggelos, M. Walton, and O. Cossairt, "Hand-guided qualitative deflectometry with a mobile device," *Opt. Express* **28**, 9027–9038 (2020).
5. H. Richard and M. Raffel, "Principle and applications of the background oriented schlieren (BOS) method," *Meas. Sci. Technol.* **12**, 1576 (2001).
6. B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel, "Time-resolved 3D capture of non-stationary gas flows," *ACM Transactions on Graph.* **27**, 132 (2008).
7. I. Ihrke, K. N. Kutulakos, H. P. Lensch, M. Magnor, and W. Heidrich, "Transparent and specular object reconstruction," in *Computer Graphics Forum*, vol. 29 (Wiley Online Library, 2010), pp. 2400–2426.
8. C. Wang, Q. Fu, X. Dun, and W. Heidrich, "Quantitative phase and intensity microscopy using snapshot white light wavefront sensing," *Sci. Reports* **9**, 1–12 (2019).
9. G. Wetzstein, R. Raskar, and W. Heidrich, "Hand-held schlieren photography with light field probes," in *IEEE International Conference on Computational Photography*, (IEEE, 2011), pp. 1–8.
10. C. Wang, X. Dun, Q. Fu, and W. Heidrich, "Ultra-high resolution coded wavefront sensor," *Opt. Express* **25**, 13736–13746 (2017).
11. D. Wang, P. Xu, Z. Wu, X. Fu, R. Wu, M. Kong, J. Liang, B. Zhang, and R. Liang, "Simultaneous multisurface measurement of freeform refractive optics based on computer-aided deflectometry," *Optica* **7**, 1056–1064 (2020).
12. J. Ye, Z. Niu, X. Zhang, W. Wang, and M. Xu, "Simultaneous measurement of double surfaces of transparent lenses with phase measuring deflectometry," *Opt. Lasers Eng.* **137**, 106356 (2021).
13. L. R. Graves, H. Choi, W. Zhao, C. J. Oh, P. Su, T. Su, and D. W. Kim, "Model-free deflectometry for freeform optics measurement using an iterative reconstruction technique," *Opt. Lett.* **43**, 2110–2113 (2018).
14. E. Olesch, C. Faber, and G. Häusler, "Deflectometric self-calibration for arbitrary specular surfaces," in *Proc. DGao*, (2011), p. P27.
15. H. Ren, F. Gao, and X. Jiang, "Iterative optimization calibration method for stereo deflectometry," *Opt. Express* **23**, 22060–22068 (2015).
16. Y. Xu, F. Gao, Z. Zhang, and X. Jiang, "A holistic calibration method with iterative distortion compensation for stereo deflectometry," *Opt. Lasers Eng.* **106**, 111–118 (2018).
17. X. Xu, X. Zhang, Z. Niu, W. Wang, Y. Zhu, and M. Xu, "Self-calibration of in situ monoscopic deflectometric measurement in precision optical manufacturing," *Opt. Express* **27**, 7523–7536 (2019).
18. C. Sun, Y. Zheng, F. Zeng, Q. Xue, W. Dai, W. Zhao, and L. Huang, "Simultaneous measurement of displacement, pitch, and yaw angles of a cavity output mirror based on phase measuring deflectometry," *Appl. Opt.* **59**, 3270–3284 (2020).
19. L. Huang, J. Xue, B. Gao, C. McPherson, J. Beverage, and M. Idir, "Modal phase measuring deflectometry," *Opt. Express* **24**, 24649–24664 (2016).
20. H. Zhang, I. Šics, J. Ladrera, M. Llonch, J. Nicolas, and J. Campos, "Displacement-free stereoscopic phase measuring deflectometry based on phase difference minimization," *Opt. Express* **28**, 31658–31674 (2020).
21. Y.-C. Leung and L. Cai, "3D reconstruction of specular surface by combined binocular vision and zonal wavefront reconstruction," *Appl. Opt.* **59**, 8526–8539 (2020).
22. L. Huang, H. Choi, W. Zhao, L. R. Graves, and D. W. Kim, "Adaptive interferometric null testing for unknown freeform optics metrology," *Opt. Lett.* **41**, 5539–5542 (2016).
23. M. C. Knauer, C. Richter, and G. Häusler, "Measuring the refractive power with deflectometry in transmission," in *Proc. DGao*, (2008), p. A24.
24. J. Bartsch, Y. Sperling, and R. B. Bergmann, "Efficient vision ray calibration of multi-camera systems," *Opt. Express* **29**, 17125–17139 (2021).

25. A. S. Jurling and J. R. Fienup, "Applications of algorithmic differentiation to phase retrieval algorithms," *J. Opt. Soc. Am. A* **31**, 1348–1359 (2014).
26. J. Werner, M. Hillenbrand, A. Hoffmann, and S. Sinzinger, "Automatic differentiation in the optimization of imaging optical systems," *Schedae Informaticae* **21**, 169–175 (2012).
27. J.-B. Volatier, Á. Menduiña-Fernández, and M. Erhard, "Generalization of differential ray tracing by automatic differentiation of computational graphs," *J. Opt. Soc. Am. A* **34**, 1146–1151 (2017).
28. G. Côté, J.-F. Lalonde, and S. Thibault, "Extrapolating from lens design databases using deep learning," *Opt. Express* **27**, 28279–28292 (2019).
29. G. Côté, J.-F. Lalonde, and S. Thibault, "Deep learning-enabled framework for automatic lens design starting point generation," *Opt. Express* **29**, 3841–3854 (2021).
30. C. Wang, N. Chen, and W. Heidrich, "Lens design optimization by back-propagation," in *International Optical Design Conference*, (Optical Society of America, 2021), pp. JTh4A–2.
31. E. Bostan, R. Heckel, M. Chen, M. Kellman, and L. Waller, "Deep phase decoder: Self-calibrating phase microscopy with an untrained deep neural network," *Optica* **7**, 559–562 (2020).
32. S. Ghosh, Y. S. Nashed, O. Cossairt, and A. Katsaggelos, "ADP: Automatic differentiation ptychography," in *IEEE International Conference on Computational Photography*, (IEEE, 2018), pp. 1–10.
33. S. Kandel, S. Maddali, M. Allain, S. O. Hruszkewycz, C. Jacobsen, and Y. S. Nashed, "Using automatic differentiation as a general framework for ptychographic reconstruction," *Opt. Express* **27**, 18653–18672 (2019).
34. Q. Guo, H. Tang, A. Schmitz, W. Zhang, Y. Lou, A. Fix, S. Lovegrove, and H. M. Strasdat, "Raycast calibration for augmented reality HMDs with off-axis reflective combiners," in *IEEE International Conference on Computational Photography*, (IEEE, 2020), pp. 1–12.
35. J. Lyu, B. Wu, D. Lischinski, D. Cohen-Or, and H. Huang, "Differentiable refraction-tracing for mesh reconstruction of transparent objects," *ACM Transactions on Graph.* **39**, 195 (2020).
36. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.* **18**, 1–43 (2018).
37. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds. (Curran Associates, Inc., 2019), pp. 8024–8035.
38. J. Meiron, "Damped least-squares method for automatic lens design," *J. Opt. Soc. Am.* **55**, 1105–1109 (1965).
39. <https://github.com/vccimaging/DiffDeflectometry> (2021).
40. M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path," *Appl. Opt.* **41**, 7437–7444 (2002).
41. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis Mach. Intell.* **22**, 1330–1334 (2000).
42. C. Zuo, S. Feng, L. Huang, T. Tao, W. Yin, and Q. Chen, "Phase shifting algorithms for fringe projection profilometry: A review," *Opt. Lasers Eng.* **109**, 23–59 (2018).
43. I. Trumper, H. Choi, and D. W. Kim, "Instantaneous phase shifting deflectometry," *Opt. Express* **24**, 27993–28007 (2016).
44. A. Mikš and P. Pokorný, "Simple method for determination of parameters of cemented doublet," *Appl. Opt.* **55**, 5456–5458 (2016).
45. S. K. Patra, J. Bartsch, M. Kalms, and R. B. Bergmann, "Phase measurement deviations in deflectometry due to properties of technical surfaces," in *Applied Optical Metrology III*, vol. 11102 (International Society for Optics and Photonics, 2019), p. 111020Q.
46. X. Zhang, Z. Niu, J. Ye, and M. Xu, "Correction of aberration-induced phase errors in phase measuring deflectometry," *Opt. Lett.* **46**, 2047–2050 (2021).
47. J. Ye, L. Chen, X. Li, Q. Yuan, and Z. Gao, "Review of optical freeform surface representation technique and its application," *Opt. Eng.* **56**, 110901 (2017).
48. C. De Boor, *A practical guide to splines: Revised edition* (Springer-Verlag, 2001), vol. 27, pp. 109–115.