

# Polynomial Optics: A Construction Kit for Efficient Ray-Tracing of Lens Systems

Matthias B. Hullin<sup>†1</sup> Johannes Hanika<sup>2</sup> Wolfgang Heidrich<sup>1</sup>

<sup>1</sup>The University of British Columbia <sup>2</sup>Weta Digital

## Abstract

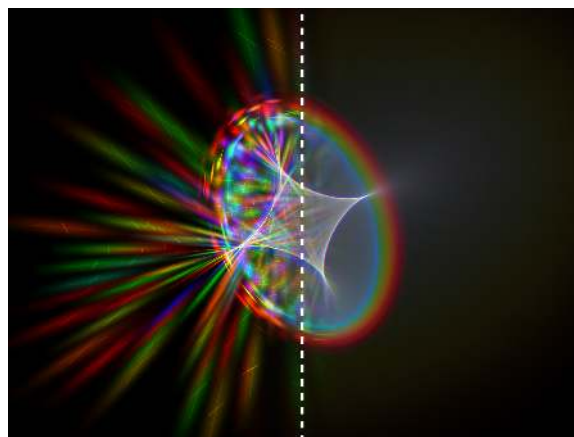
*Simulation of light transport through lens systems plays an important role in graphics. While basic imaging properties can be conveniently derived from linear models (like ABCD matrices), these approximations fail to describe nonlinear effects and aberrations that arise in real optics. Such effects can be computed by proper ray tracing, for which, however, finding suitable sampling and filtering strategies is often not a trivial task. Inspired by aberration theory, which describes the deviation from the linear ray transfer in terms of wavefront distortions, we propose a ray-space formulation for nonlinear effects. In particular, we approximate the analytical solution to the ray tracing problem by means of a Taylor expansion in the ray parameters. This representation enables a construction-kit approach to complex optical systems in the spirit of matrix optics. It is also very simple to evaluate, which allows for efficient execution on CPU and GPU alike, including the computation of mixed derivatives of any order. We evaluate fidelity and performance of our polynomial model, and show applications in high-quality offline rendering and at interactive frame rates.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

## 1. Motivation

The simulation of lens systems is a crucial ingredient to realistic image synthesis, and an active topic in graphics research [Kes08, LES10]. Most such simulations are based on the ray tracing approach, which is a simple concept but not without its flaws: Ray-surface intersection tests are computationally expensive and can become unstable for systems with high numerical aperture, especially when evaluated on a sparse grid [HESL11]. Naïve ray tracing of complex optical systems typically results in a large number of wasted samples [SDHL11]. Modern rendering schemes therefore employ sophisticated sampling and filtering strategies, to make efficient use of the compute budget. Those techniques are often based on ray differentials to obtain information about the neighborhood around a given light path [Ige99].

In this paper, we introduce a novel approach to ray tracing of optical systems. Starting with the analytical solution of the ray-surface intersection problem, we translate it with a Taylor approximation up to a desired degree. Our technique can be regarded as a ray-space interpretation of aberration theory. Compared to previous approaches, it has the following benefits:



**Figure 1:** *Caustic of light falling through a lens (Edmund Optics, article # NT49-291) where image formation is dominated by ray optics. For each wavelength, we determine a polynomial system of degree 3, mapping light rays falling through the entrance aperture to a location in the film plane. The Jacobian of the mapping (crossed lines) and the filter derived from it are shown in the left half. Analytical derivatives are useful in various stages of a rendering pipeline.*

<sup>†</sup> hullin@cs.ubc.ca

- Polynomials are a generalization of ray transfer matrices to higher orders, and they can be used and combined in a similarly intuitive construction-kit manner.
- Polynomials are cheap to evaluate. Truncation of higher-degree terms keeps the computation effort constant with respect to the complexity of the optical system. The near-zero memory footprint for the polynomial coefficients makes the technique ideal for GPU implementation.
- Polynomials are mathematically well-behaved, and computation of analytical derivatives is straightforward. This does not only enable efficient filtering in density estimation problems, but also lends itself to an elegant solution of inverse problems that occur in path tracing scenarios.
- Optical systems abstracted through polynomial mappings can easily be integrated in existing ray tracers.

In this paper, we lay the theoretical foundations of polynomial optics, and relate it to linear transport matrices. We highlight possible application scenarios for high-quality rendering and computational photography. Full source code and usage examples are provided as supplemental material on the project website, <http://www.cs.ubc.ca/labs/imager/tr/2012/PolynomialOptics/>.

## 2. Related work

The simplest way to describe the propagation of light rays through an optical system is in terms of ray transfer matrices, also known as *ABCD* matrices. They can be used to characterize light transport in the proximity of an optical axis, as well as idealized thin lenses and similar components. Because of the linear formalism, multiple components and transport operators can be combined into a single *system matrix* describing the entire assembly.

Building upon Gauss' development of such a linear model [Gau41], Seidel went on to identify and characterize *aberrations*, i.e. higher order deviations from the linear mapping [Sei57]. Optics researchers have since been working on ever more sophisticated descriptions of the imaging properties of optical systems.

The most precise methods today are rooted in Fresnel-Huygens theory, and describe aberrations in terms of wavefronts or phase surfaces. The propagation of such wavefronts through space is expensive to compute since it involves the evaluation of integrals over a dense set of elementary waves. Only recently, first attempts have been made to marry expressive aberration models with matrix optics to describe their evolution along an optical path [DCC\*09,HNK\*11,EBM\*11]. Polynomial bases allow for a thorough mathematical treatment, the most prominent example being Zernike's orthonormal basis of wavefront polynomials over the unit circle [BW99]. Thanks to a closed-form expression for the 2D Fourier transform [Tan77], Zernike polynomials are widely used in astronomy and the semiconductor industry where the point spread function is limited by diffraction, not by geometric aberrations.

In this work, we also use a polynomial basis, however, instead of wavefronts our technique directly operates in ray space, and we assume that light transport is entirely governed by geometric optics. We thus cater to the fact that for the vast majority of computer graphics and machine vision applications, geometric optics is usually not only sufficient, but also easier and more efficient to use than a full wave formulation. In practice, wave effects are usually neglected or mimicked using simple Fourier relations [RIF\*09,HESL11].

As far as the camera model is concerned, simple linear systems (such as camera matrices) capture the image formation process better than just pinhole models, and can in particular model a finite depth of field. Other aspects of real lens systems, such as radial distortion, spherical or chromatic aberrations, etc. are not captured by linear systems. Their unified treatment requires some form of ray-tracing, as has been used for offline [KMH95,LSS98] and real-time rendering [HSS97,LES10].

With this paper, we seek to combine the ease of use and the algebraic elegance of *ABCD* matrices with a close approximation of aberrations through optical systems. The use of simple multivariate polynomials allows for cheap evaluation and computation of derivatives that are extremely useful for differential rendering techniques [Ige99]. This makes polynomial optics an interesting alternative to traditional ray tracing.

## 3. Model

**Ray geometry.** Like in the paraxial approximation used in matrix optics, we assume a coordinate frame that is defined by an optical axis  $\hat{\mathbf{z}}$ , with vectors  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{y}}$  completing an orthonormal basis. Any ray  $\mathbf{r}$  in 3D space is fully defined by a point  $\mathbf{p} = [p_x, p_y, p_z]$  and a unit direction vector  $\mathbf{d} = [d_x, d_y, d_z]$ ,  $|\mathbf{d}| = 1$ , which we bundle in the *ray vector*  $\mathbf{r}$ :

$$\mathbf{r} = [p_x, p_y, p_z, d_x, d_y, d_z]^T \quad (1)$$

Assuming that the direction component  $d_z$  is positive, we can obtain it from  $d_x$  and  $d_y$  as

$$d_z = \sqrt{1 - d_x^2 - d_y^2} \quad (2)$$

Any ray passing through the plane  $z = z_0$  is therefore fully characterized by four variables,  $p_x, p_y, d_x$  and  $d_y$ , resulting in a *reduced* ray vector  $\tilde{\mathbf{r}}$  with respect to this plane:

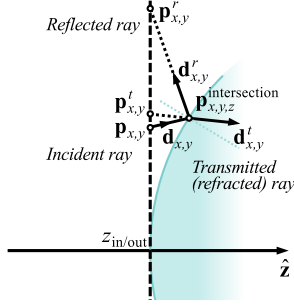
$$\tilde{\mathbf{r}} = [\tilde{p}_x, \tilde{p}_y, d_x, d_y]^T, \quad \text{where} \quad (3)$$

$$\tilde{p}_x = p_x + d_x/d_z(z_0 - p_z) \quad \text{and} \quad (4)$$

$$\tilde{p}_y = p_y + d_y/d_z(z_0 - p_z). \quad (5)$$

As a special case, the optical axis itself has neither displacement nor direction components orthogonal to  $\hat{\mathbf{z}}$ . Hence, its reduced ray vector is

$$\tilde{\mathbf{z}} = [0, 0, 0]^T. \quad (6)$$

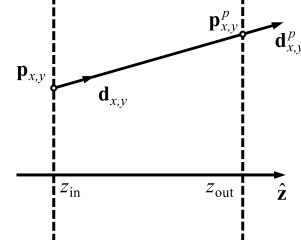


**Figure 2:** Intersection of a ray with a spherical interface (blue) centered on the optical axis  $\hat{z}$ . Position and direction vectors before ( $\mathbf{p}, \mathbf{d}$ ) and after transmission ( $\mathbf{p}^t, \mathbf{d}^t$ ) and reflection ( $\mathbf{p}^r, \mathbf{d}^r$ ) are specified with respect to the input/output plane (dashed vertical line). The virtual intersection points  $\mathbf{p}^t, \mathbf{p}^r$  in that plane are obtained by translating the true 3D intersection point  $\mathbf{p}^{\text{intersection}}$  along the respective direction vectors (dotted lines).

We denote the set of all reduced ray vectors as  $\mathbb{V} = \mathbb{R}^2 \times \mathbb{D}_2$ , where  $\mathbb{D}_2$  is the set of all pairs  $[d_x, d_y]$  within the unit circle of  $\mathbb{R}^2$  (in order for Equation 2 to have a real solution). Since all of the following discussion relates to reduced ray vectors, we drop the tilde for simplicity of notation.

We now let our ray propagate through an optical system consisting of  $N$  spherical surfaces that are interfaces between homogeneous optical media. These lens surfaces are counted following their order of intersection with the optical axis in  $+\hat{z}$  direction, with intersection coordinates  $z_1 \dots z_N$ . Just as with transfer matrices, a key insight is that even for curved surfaces such as spheres, we can treat all interactions in terms of reduced ray vectors in designated planes that are perpendicular to the optical axis. Unlike traditional matrix optics, however, we do not start by making the paraxial assumption of small angles ( $x \approx \sin x \approx \tan x$ ;  $\cos x \approx 1$ ). Instead, we initially derive the ray propagation by solving the 3D ray-surface intersection, reflection and refraction analytically. We use the symbolic algebra package “Maple” to obtain the corresponding expressions. With typically half a page per equation, they are too bulky for print and, in addition, do not offer much theoretical insight. Later on, we use their Taylor expansions. Note that in principle, the optical surfaces can have any shape as long as their intersection with a ray has an analytical solution. In the following, to simplify our argumentation, we assume either rotational or reflection symmetry, which holds for most optical elements.

Let us consider the refraction of a ray that is transmitted through the spherical surface with index  $i$ . We choose the same plane for input and output, placed at the intersection point with the optical axis,  $z = z_i$ . The rays before and after refraction are given as reduced ray vectors with respect to that plane. Although the intersection point  $\mathbf{p}^{\text{intersection}}$ , in general, lies outside the plane, it is straightforward to project it back along the direction of the refracted ray, obtaining the



**Figure 3:** Free-space propagation of a ray from plane  $z_{in}$  to plane  $z_{out}$ . The direction vector remains unchanged ( $\mathbf{d}^p = \mathbf{d}$ ), but the position is translated from  $\mathbf{p}$  to  $\mathbf{p}^p$ .

virtual intersection point  $\mathbf{p}^r$  as seen from the output plane (Figure 2).

We distinguish between *interfaces*, where input and output planes are identical, and *propagation* along the optical axis in free space or within the medium (Figure 3). Our argumentation follows the deterministic ray tracing scheme proposed by [KMH95]. Given a physically plausible lens system where the optical surfaces do not intersect each other, and assuming the absence of total internal reflection, all rays experience the same sequence of interfaces and propagations. It is therefore technically not necessary to *test* for ray-surface intersection, if the resulting ray geometry can be obtained without that test.

**Ray mapping.** The effect of an optical system on incoming rays, as described in the previous section, can be abstracted as a mapping, or function  $\mathbf{f}: \mathbb{V} \rightarrow \mathbb{V}$ :

$$\begin{bmatrix} r'_x \\ r'_y \\ d'_x \\ d'_y \end{bmatrix} = \mathbf{r}' = \mathbf{f}(\mathbf{r}) = \begin{bmatrix} f_{r_x} \\ f_{r_y} \\ f_{d_x} \\ f_{d_y} \end{bmatrix} (r_x, r_y, d_x, d_y) \quad (7)$$

Assuming that all optical surfaces in our system are analytic,  $\mathbf{f}$  too will be analytic and hence smooth on a domain  $\mathbb{V}_{\text{center}} \subseteq \mathbb{V}^\dagger$ . We exploit these two facts by expanding Eq. 7 in a multivariate Taylor series:

$$\begin{bmatrix} f_{r_x} \\ f_{r_y} \\ f_{d_x} \\ f_{d_y} \end{bmatrix} (r_x, r_y, d_x, d_y) = \sum_{a=0}^{\infty} \sum_{b=0}^{\infty} \sum_{c=0}^{\infty} \sum_{d=0}^{\infty} \sum_{a+b+c+d \geq 1} \begin{bmatrix} \alpha_{r_x} \\ \alpha_{r_y} \\ \alpha_{d_x} \\ \alpha_{d_y} \end{bmatrix}_{(a,b,c,d)} \cdot r_x^a \cdot r_y^b \cdot d_x^c \cdot d_y^d, \quad (8)$$

i.e.,  $\mathbf{r}'$  is given by a system of four equations that are each a power series in the four components of  $\mathbf{r}$ . The *total degree* of any term in such a series is the sum of all its exponents,  $a + b + c + d$ .

$\dagger$   $\mathbf{f}$  will not necessarily be smooth on all of  $\mathbb{V}$ , for instance at the critical angle of total internal reflection (i.e., when there is no real solution for Snell’s law at a refractive surface), or when a ray no longer intersects the spherical surface. Our model does not cover such discontinuities, but extrapolates the mapping from  $\mathbb{V}_{\text{center}}$  to the rest of  $\mathbb{V}$ .

As with functions of a single variable, a mapping  $\mathbf{f}$  with a convergent Taylor series can be approximated by truncating it to a system of polynomials of a desired total degree, which we define as the overall total degree of our mapping, denoted as  $\text{deg}(\mathbf{f})$ . For the solutions to our geometric problems, the mappings are well-behaved in the sense that their Taylor series converge over the entire domain for which an analytical solution exists.

Table 1 contains an overview of the nonzero term count for a selection of optical elements. Notably, symmetric systems only produce terms with an odd total degree. The Taylor coefficients depend on the parameters (radius, refractive indices); we refer the interested reader to our C++ library for more information. In Section 4, we demonstrate how various polynomial terms affect the imaging properties of a system.

Case		Term degree				
		1	3	5	7	9
(a)	$r'_x, r'_y$	1	4	18	46	92
	$d'_x, d'_y$	2	7	23	53	101
(b)	$r'_x, r'_y$	1	4	16	44	90
	$d'_x, d'_y$	2	7	15	40	83
(c)	$r'_x$	1	1	4	9	16
	$r'_y$	1	2	6	12	20
	$d'_x$	1	0	0	0	0
	$d'_y$	2	4	9	16	25
(d)	$r'_x$	1	1	4	9	16
	$r'_y$	1	2	5	10	17
	$d'_x$	1	0	0	0	0
	$d'_y$	2	4	7	12	19
(e)	$r'_x, r'_y$	2	2	3	4	5
	$d'_x, d'_y$	1	0	0	0	0

**Table 1:** Number of nonzero Taylor coefficients for (a) refraction and (b) reflection at a spherical surface, (c) refraction and (d) reflection at an cylindrical surface oriented in  $\hat{\mathbf{x}}$  direction, (e) free-space propagation. The anamorphic lens flare in Figure 9 is the only example in this paper that uses cylindrical elements.

**A construction-kit approach.** The ray transfer matrix is a common linear approximation to the mapping performed by an optical system, allowing for many useful calculations such as the focal length of a lens system, and stability of optical resonators [BW99]. We obtain it as a special case by only allowing degree-1 terms in Eq. 8. One of the beautiful properties of the matrix formalism is that it can serve as a construction kit for arbitrary combinations of refractive and reflective elements for which the linear approximation holds: the mapping of the composite system is then simply the product of the matrices of the individual elements.

Likewise, using our polynomial model we can concatenate optical elements by plugging the output of one into the input of the other:

$$(\mathbf{g} \circ \mathbf{f})(\mathbf{r}) = \mathbf{g}(\mathbf{f}(\mathbf{r})) \quad (9)$$

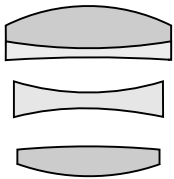
It is important to note that every such step roughly multiplies the number of terms of the respective polynomial systems. We therefore initially determine a maximum polynomial degree  $n$  up to which the optical system shall be computed, and after each concatenation operation truncate the system back to that degree. This prevents the exponential increase in complexity, making the assembly of complex optical systems tractable. At the same time, the result is identical to the full polynomial expansion followed by a single truncation:

$$\text{trunc}(\mathbf{g} \circ \mathbf{f}, n) = \text{trunc}(\mathbf{g} \circ \text{trunc}(\mathbf{f}, n), n) \quad (10)$$

This follows from the fact that, when a polynomial is substituted into another polynomial, its total degree can only increase. Terms with a degree higher than the maximum degree of interest  $n$  can therefore be discarded at once.

**Scalar modulation.** The polynomial approach to describing the geometric mapping of light rays can to some extent also be applied to scalar functions, most prominently the attenuation of the ray. An example containing both geometric mapping and attenuation is sketched in the supplemental document to this paper.

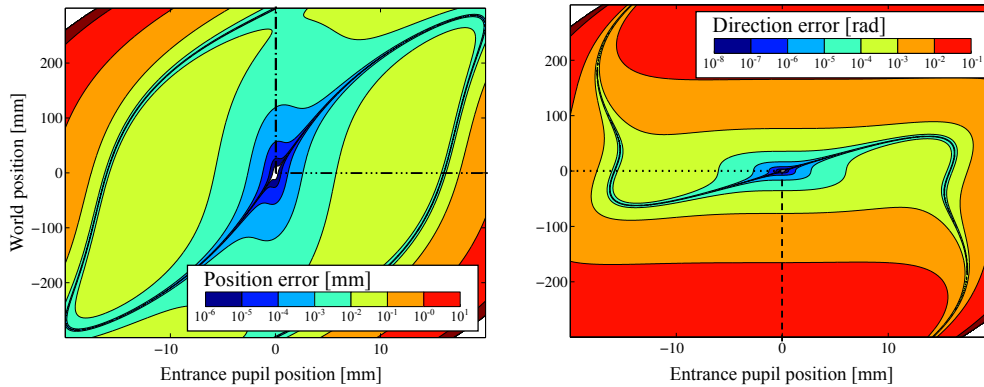
**Evaluation.** To compute the value of a polynomial for a given set of variable values, the straightforward approach is to store all coefficients for every power of the input variables and walk them in for-loops, using multiply-add to obtain the result. Interpreting a polynomial this way can be costly to compute, since it introduces branches and does not re-use monomial sub-terms that may occur several times. The alternative is to compile the polynomial down to native machine code, using the coefficients as constants, and to evaluate this for many data points (light position, aperture size, and optionally other dynamic parameters such as zoom setting and wavelength) after that.

radius[mm]	thick.[mm]	material	
42.97	9.80	LAK9	
-115.33	2.10	LLF7	
306.84	4.16	air	
-59.06	1.87	SF7	
40.93	10.64	air	
183.92	7.05	LAK9	
-48.91	79.83	air	

**Figure 4:** We compare our polynomial approximation to an analytical ray tracing solution for a 100 mm,  $f/2.8$  tessar design ("brendel", US Patent 2854889). At a distance of 1500 mm from the entrance pupil, the field-of-view diagonal of this lens (on full-format film) is 600 mm.

#### 4. Characterization

**Comparison against analytical solution.** In order to validate the performance of polynomial optics with respect to true ray tracing, we evaluate a test scenario in different



**Figure 5:** Error of matrix optics for the optical system from Figure 4. Focusing at a distance of 1500 mm, we connect points in the object plane (vertical axis) with all points on the entrance pupil (horizontal axis), propagate the corresponding rays to the image plane, and compute the error with respect to the analytical solution. Left column: intersection point. Right column: ray direction. The dotted-dashed lines mark the cutting planes for the profiles plotted in Figure 6.

polynomial degrees, and compare the resulting ray mapping against an analytical solution. We choose the lens system shown in Figure 4, a historic tessar design. Lenses of this complexity typically compensate for some nonlinear aberrations, which makes this an interesting candidate to compare against. As reference, we plot the reconstruction error of the corresponding ABCD matrix (Figure 5). Comparison with polynomial degrees 3 and 5 is provided for selected cuts through the variable space in Figure 6.

Since the Taylor expansion is centered around the optical axis, it performs best around the center and worse in the extreme periphery. Unsurprisingly, polynomials of degree 3 or 5 consistently result in a smaller reconstruction error than the linear model (degree 1). Only for far off-axis rays, the benefits vanish, and the quality of higher-degree approximations drops below the one of the linear model. However, we note that such rays are outside the physically plausible range of the lens system and would be blocked by the barrel, which was neglected in our simulation.

**Aberrations.** In accordance with optics literature, we find that the “textbook aberrations” are well described with terms of degree 3. In axisymmetric systems, these aberrations appear as groups of two or three terms (Figure 7). In Figure 8, we illustrate the effect of nonlinear terms on the resulting rays. It is interesting to note an apparent shift of the focal plane that is in fact caused by spherical aberration. A smaller entrance pupil blocks those aberrated rays and moves the effective focus back towards the nominal focal plane.

## 5. Application examples and results

In the following, we demonstrate a few potential applications of polynomial optics. In all cases, we used a truncation degree of 3. In our experience, an approximation up to degree

5 only changes the outcome marginally, but significantly increases the computational effort (count and degree of terms).

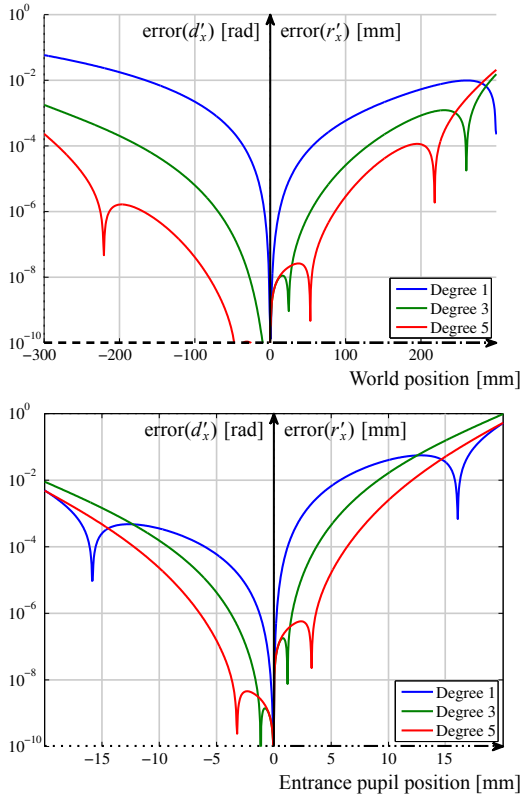
**Image processing.** Simple lens systems have a characteristic defocus behavior that can be used to achieve interesting effects. In Figure 13, we show an HDR photo of a night scene that is re-traced through an achromatic doublet. Thanks to the far distance of the scene (several kilometers), it is safe to abstract it as a plane.

**Flare computation.** We now demonstrate that polynomial optics can be used to efficiently render lens flare. Starting with a lens prescription, we first enumerate all possible light paths involving two reflection events, and determine a polynomial system for each of them. The resulting polynomial expressions are then used to generate code, which is compiled just-in-time and executed using the OpenCL framework [ope11]. This has the advantage that we are able to compile both for the GPU and to vectorized CPU code.

The compute kernel draws a random sample from the entrance pupil, then uses the hard-coded polynomials to determine the image plane positions of the ghosts for this sample. The sample is then splatted additively into a shared buffer.

To include occlusion of rays inside the lens housing due to the aperture, we evaluate one more polynomial for each ray to find the position where it passes the iris. Since we only evaluate flares caused by two reflection events, this results in a maximum of three extra polynomials (pass the aperture on the way in, reflect, pass it again, reflect back from the other side and pass it one final time). The ray is then potentially discarded based on a point/polygon test.

Note that we do not apply any of the numerous high-level acceleration strategies used in prior work; in particular we do not perform any ray planning step or interpolation. Still, our approach delivers high-quality renderings

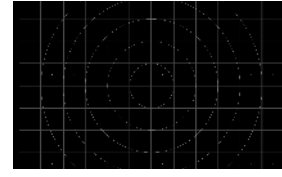


**Figure 6:** Absolute error of degree-1 to degree-5 approximations with respect to analytical ray tracing of the optical system from Figure 4. The horizontal axes mark the  $x$  coordinate in the object and entrance pupil planes, respectively, with dash-dot patterns corresponding to the cuts marked in Figure 5. The vertical axis is shared between the errors for direction ( $d'_x$ ) and ray position ( $r'_x$ ) after propagation to the image plane.

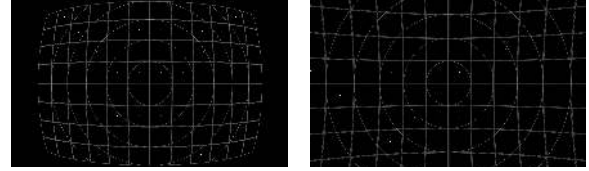
(Figure 9) and near-realtime performance. The achieved rendering times are in the order of 1 second (“*canon*”), 0.4 seconds (“*itoh*”), and 4 seconds (“*canon-anamorphic*”), on an NVIDIA GTX 550 Ti chipset. It is therefore to be expected that future refinements of the technique will yield another substantial speed-up.

If compile times are a concern (for the anamorphic example with 861 flares, code generation and compilation takes about 8 minutes on a single CPU core), they can be further optimized. For instance, LLVM [llv11] can be leveraged to directly produce a vectorized intermediate representation of the polynomial code (via `llvm::IRBuilder`), which can then be compiled and executed directly in memory (i.e. without parsing a source file). Also OpenCL compilers usually assume the kernels are small and can thus be optimized using very expensive techniques. These can be very finely controlled when manually compiling in memory.

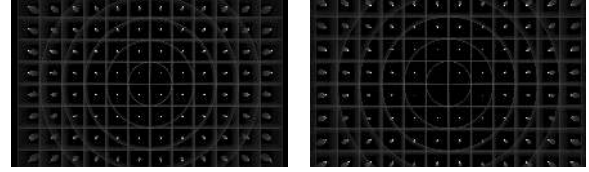
(a) Undistorted image:  $r'_x = -r_x$



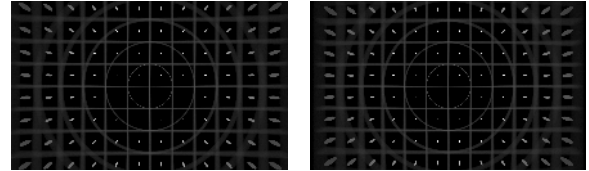
(b) Barrel/Pincushion:  $r'_x = -r_x \pm \alpha_{\text{radial}} \cdot r_x(r_x^2 + r_y^2)$



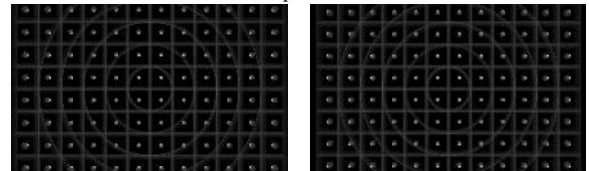
(c) Coma:  $r'_x = -r_x \pm \alpha_{\text{coma}} \cdot (3r_x d_x^2 + 2r_y d_x d_y + r_x d_y^2)$



(d) Curvature:  $r'_x = -r_x \pm \alpha_{\text{curv}} \cdot (3r_x^2 d_x + 2r_x r_y d_y + r_y^2 d_x)$

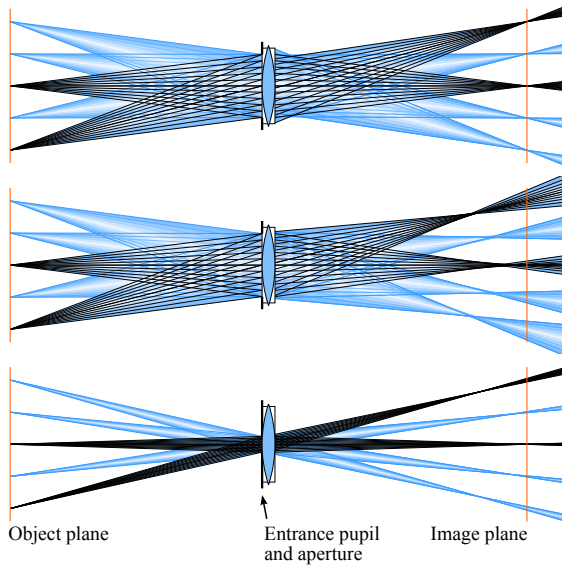


(e) Spherical:  $r'_x = -r_x \pm \alpha_{\text{sph}} \cdot d_x(d_x^2 + d_y^2)$



**Figure 7:** In a degree-3 polynomial produced by an axially symmetrical lens system, we identify four groups of cubic terms that form radially symmetric image distortions. We provide the formula that maps the ray coordinate tuple  $(r_x, r_y, d_x, d_y)$  to the image plane position  $r'_x$ . Note that Seidel’s aberration definitions (spherical aberration, coma, field curvature, defocus) do not map exactly to our polynomial basis; however, these phenomena do not occur in isolated form in real optics either.

In addition to physically motivated lens flare driven by an actual lens prescription, we can also directly start with the groups of terms identified in Figure 7. Random coefficients result in beautiful caustic-like effects that share many similarities with real lens flare, but do not correspond to any plausible lens system (Figure 10).



**Figure 8:** Polynomial abstraction of a thick biconvex lens (100 mm,  $f/2.0$ ) in a 1:1 imaging setting. Degree of polynomial from top to bottom: 1, 3, 3 stopped down to  $f/8.0$ . The sketched lens stands for the “black box” polynomial system. Note how the addition of cubic terms introduces spherical aberration (causing the beam waist to fall before the first-order focal plane), and field curvature.

**3D rendering.** Our polynomial model describes nothing other than the mapping of an incoming to an outgoing ray and hence can trivially replace any camera in a renderer. A basic depth-of-field example using the “*itoh*” zoom lens is shown in Figure 11.

**Using derivatives for filtering.** In Figure 1, we demonstrate how a polynomial representation can be used to obtain high-quality renderings of caustics and similar optical phenomena. For a glass lens, we determined the polynomial system and used a submatrix of its Jacobian (namely, the derivatives of the image plane position with respect to the aperture position) in order to determine the size and orientation of a Gaussian filter kernel. This efficiently reduces sampling noise in smooth regions while preserving high-frequency details of the caustic.

Additional parameters, such as the wavelength or lens settings, can be included as variables in the system. Thus, it becomes possible to estimate good filter kernels even for density estimation problems with higher-dimensional parameter spaces, albeit at an increased number of polynomial terms.

This kind of filtering is even more important when only a discrete subset of ray samples is available. Figure 12 shows such a difficult case where we used a light field dataset from the Stanford Light Field Archive [Sta] as input data. The Ja-

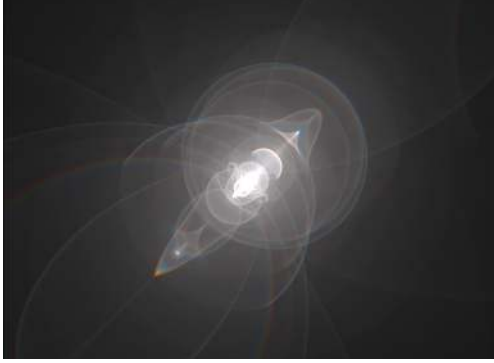


**Figure 9:** Flare renderings of three different lens systems generated using polynomial optics. In this order: “*canon*” and “*itoh*” zoom lenses also used in [HESL11] (34 and 19 surfaces, respectively), “*canon-anamorphic*” (42 surfaces, US Patent 3990785). Only indirect light was rendered, and we used the incorrect assumption that all surfaces reflect the same amount of light.

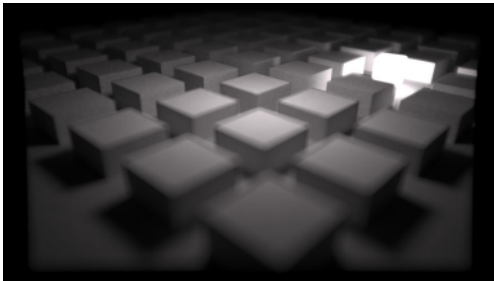
cobian would have to depend on the distance of the individual features in the scene to obtain a proper filter kernel and interpolate between the sparse samples. Even if we had such data, this feature that is currently not provided by our library of optical elements. As a workaround for the time being, we propose use of the chain rule, along with the required geometric relations in object space, in order to obtain derivatives that are corrected for scene depth.

## 6. Conclusion

In this paper, we have introduced a novel approach to computing the propagation of light rays through optical systems. Unlike brute-force ray tracing, once the polynomial systems have been set up, the computational complexity of our technique does not depend on the size (number of optical surfaces) of the optical system, but only on the polynomial degree to which we approximate it. When exceeding the phys-



**Figure 10:** “Fake flare” rendering, where each ghost is obtained by a random linear combination of the nonlinear term groups identified in Figure 7, with a bit of spectral variation added to the coefficients.



**Figure 11:** Simple 3D depth of field rendering using particle tracing. The endpoints of light paths are splatted to the sensor by connecting to the entrance pupil. As our implementation of the interface between renderer and lens simulator ignores visibility between these points, there are some dark edges at depth discontinuities. Nevertheless, the image shows some interesting softness and blooming around the light source, as well as subtle chromatic aberrations.



**Figure 12:** “Failure case”: A dataset from the Stanford Light Field Archive as seen through a virtual cheap lens with radial distortion and chromatic fringing. Note how the specular highlights are not blurred correctly in the peripheral regions – additional depth information per ray would be required to determine the appropriate filter kernel.

ical bounds of the system, the value of these polynomials tends to overshoot but is still well-defined. Furthermore, it is trivial to obtain first-order and higher derivatives with respect to all input variables, allowing for simple filtering of density estimation tasks, as well as bidirectional tracing of light paths in stochastic rendering.

While a higher polynomial degree naturally approximates the exact behavior of the lens better, it is sometimes hard to estimate how many terms are needed to capture a given optical system. In our experience, the majority of aberrations and effects are captured well by linear and cubic terms. More sophisticated optics are typically designed to eliminate aberrations, so a description of the residual artifacts in their point spread function may occasionally require a higher polynomial degree.

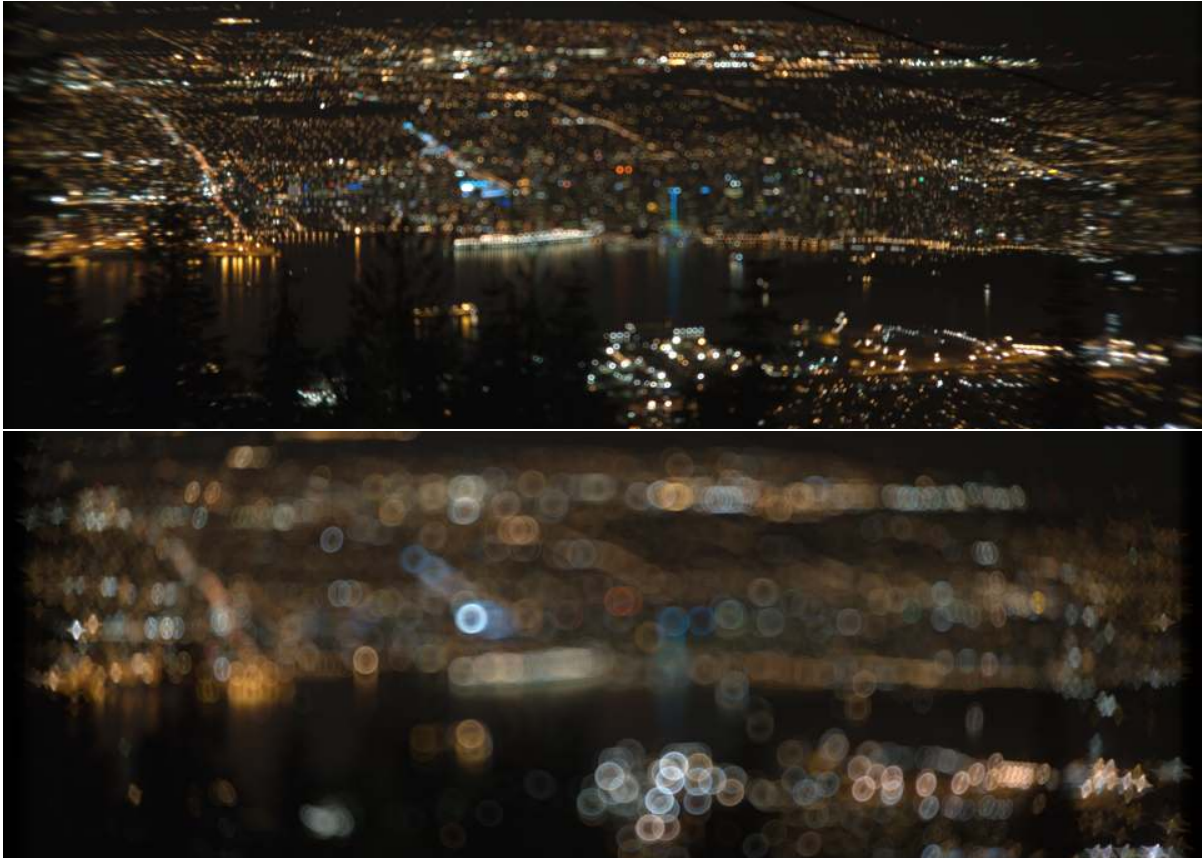
Since the ray mapping is a unique function, our method cannot account for certain optical effects, in particular stochastic scattering processes and diffraction. Even so, we find that many interesting phenomena from the realm of geometric optics can be described using the model. With this foundation, we hope to inspire others to experiment with nonlinear lens models and explore the possibilities of polynomial optics.

We thank Mushfiqur Rouf for his help with the light field example. Wolfgang Heidrich is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and a Dolby Research Chair. We also thank Joe Letteri and Sebastian Sylwan for supporting the work within Weta Digital Ltd.

## References

- [BW99] BORN M., WOLF E.: *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th Edition)*, 7th ed. Cambridge University Press, October 1999. 2, 4
- [DCC\*09] DAI G., CAMPBELL C. E., CHEN L., ZHAO H., CHERNYAK D.: Wavefront propagation from one plane to another with the use of zernike polynomials and taylor monomials. *Appl. Opt.* 48, 3 (Jan 2009), 477–488. 2
- [EBM\*11] ESSER G., BECKEN W., MÜLLER W., BAUMBACH P., ARASA J., UTTENWEILER D.: Derivation of the propagation equations for higher order aberrations of local wavefronts. *J. Opt. Soc. Am. A* 28, 12 (Dec 2011), 2442–2458. 2
- [Gau41] GAUSS C.: *Dioptrische Untersuchungen*. Dieterich, 1841. 2
- [HESL11] HULLIN M. B., EISEMANN E., SEIDEL H.-P., LEE S.: Physically-based real-time lens flare rendering. *ACM Trans. Graph. (Proc. SIGGRAPH 2011)* 30, 4 (2011), 108:1–108:9. 1, 2, 7
- [HNK\*11] HILLMAN L. W., NARAYANKAR R. A., KILARU K., SPRADLEY K. D., GREGORY D. A.: Transformations of aberrations in optical systems. *J. Opt. Soc. Am. A* 28, 9 (Sep 2011), 1820–1827. 2
- [HNS97] HEIDRICH W., SLUSALLEK P., SEIDEL H.-P.: An image-based model for realistic lens systems in interactive computer graphics. In *Graphics Interface* (May 1997), Davis W., Mantei M., Klassen V., (Eds.), pp. 68–75. 2





**Figure 13:** A HDR photo of a city at night is imaged through a virtual achromatic doublet (Edmund Optics #NT32-921), adding aberrations as the aperture size is increased (top). Note the subtle color variation within each pattern, originating from the spectral dispersion of the optical system. Deliberate defocus turns the point lights into fascinating patterns (bottom).

- [Ige99] IGEHY H.: Tracing ray differentials. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 179–186. 1, 2
- [Kes08] KESHMIRIAN A.: *A physically-based approach for lens flare simulation*. Master's thesis, University of California, San Diego, 2008. 1
- [KMH95] KOLB C., MITCHELL D., HANRAHAN P.: A realistic camera model for computer graphics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 317–324. 2, 3
- [LES10] LEE S., EISEMANN E., SEIDEL H.-P.: Real-Time Lens Blur Effects and Focus Control. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'10)* 29, 4 (2010), 65:1–7. 1, 2
- [llv11] LLVM compiler infrastructure 3.0. <http://llvm.org/>, 2011. 6
- [LSS98] LOOS J., SLUSALLEK P., SEIDEL H.-P.: Using wavefront tracing for the visualization and optimization of progressive lenses. *Computer Graphics Forum* 17, 3 (1998), 255–265. 2
- [ope11] Khronos group OpenCL 1.2 specification. <http://www.khronos.org/registry/cl/>, 2011. 5
- [RIF\*09] RITSCHEL T., IHRKE M., FRISVAD J. R., COPPENS J., MYSZKOWSKI K., SEIDEL H.-P.: Temporal Glare: Real-Time Dynamic Simulation of the Scattering in the Human Eye. In *Computer Graphics Forum (Proc. Eurographics)* (2009). 2
- [SDHL11] STEINERT B., DAMMERTZ H., HANIKA J., LENSCH H. P. A.: General spectral camera lens simulation. In *Computer Graphics Forum* (2011), vol. 30, pp. 1643–1654. 1
- [Sei57] SEIDEL L.: *Über die Theorie der Fehler, mit welchen die durch optische Instrumente gesehenen Bilder behaftet sind, und über die mathematischen Bedingungen ihrer Aufhebung*. Abhandlungen der Naturwissenschaftlich-Technischen Commission bei der Königl. Bayerischen Akademie der Wissenschaften in München. Cotta, 1857. 2
- [Sta] STANFORD UNIVERSITY: The stanford light field archive. <http://lightfield.stanford.edu/>. 7
- [Tan77] TANGO W. J.: The circle polynomials of Zernike and their application in optics. *Applied Physics* 13 (Aug. 1977), 327–332. 2