Adaptive sparse polynomial regression for camera lens simulation

Article *in* The Visual Computer · May 2017

DOI: 10.1007/s00371-017-1402-9

2 authors, including:

Quan Zheng
Chinese Academy of Sciences
8 PUBLICATIONS **6** CITATIONS

SEE PROFILE

The Visual Computer, June 2017, Volume 33, Issue 6

Adaptive sparse polynomial regression for camera lens simulation

Quan Zheng · Changwen Zheng

Abstract Lens effects are crucial visual elements in the synthetic imagery, but rendering lens effects with complex full lens models is time-consuming. This paper proposes a polynomial regression based approach for constructing a sparse and accurate polynomial lens model. Terms of a polynomial are built adaptively in a bottom-up approach. Depending on the distribution of aberrations, this approach partitions the light field and builds separate polynomial models for local light fields. A line pupil–based sampling method is presented to accelerate the generation of camera rays. In addition, a new Monte Carlo estimator is derived to support general Monte Carlo rendering. Experiments show that this approach significantly reduces the time cost of constructing a polynomial lens model in comparison to state-of-the-art methods, while achieving high imaging accuracy.

Keywords Sparse polynomial regression · Camera lens simulation · Lens effects · Photorealistic rendering

1 Introduction

Photo-realistic rendering has achieved great advances in the past decades. Nowadays, we can synthesize digital images of forests, sunset, rainbow and hair with unprecedented realism. However, simulating plausible camera lens related phenomena remains a challenging problem. When it comes to advanced defocus blur, aberrations and distortions, detailed structure of real camera lenses has to be considered. A large amount of computation is required for accurately coping with all lens elements of a complex camera lens.

Polynomial optics [1] provides a constructive kit to build a polynomial system for describing the optical properties of a real camera lens. Polynomial fitting [2] is later introduced to improve the imaging precision. Recently, sparse high-degree polynomial (SHDP) [3] is proposed to select significant terms from a bulk of candidate terms. While it can find the best combination of polynomial terms within a predefined degree, it requires considerable time if the predefined degree is set to a large value or the number of samples is large.

In this paper, we propose a novel approach to adaptively construct a sparse and accurate polynomial lens model in a machine learning manner. The construction process starts from basic low-degree terms, and it automatically builds necessary terms. We design heuristic complication operators that build new terms based on existing terms, and we also develop simplification operators that remove less significant terms to constrain the number of terms. Both theoretical and empirical comparisons show that our method behaves more efficient at constructing a polynomial regression model.

In addition, based on the observation that lens aberrations vary with respect to off-axis distances, we thus exploit the local coherence of a light field and partition the input light field into local sub-regions. Then we separately construct polynomial systems for sub-regions.

Rays traversing in a complex camera lens can be blocked by obstacles like stops or the lens barrel, without exiting the camera lens. To increase the survival rates of rays, we present a method for sampling

(✉)Quan Zheng[1,2] · Changwen Zheng[1]
quan.zheng@outlook.com; cwzheng@ieee.org
[1] Science and Technology on Integrated Information System Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
(*Authors' preprint*)

camera rays based on precomputed line pupils. Line pupil–based sampling enables fast generation of camera rays at runtime. Besides, we derive new Monte Carlo estimator to support the combination of general Monte Carlo rendering methods with the proposed polynomial regression model.

2 Previous work

Simplified lens models. Pinhole camera is the simplest model for rendering. Thin lens model [4] simulates camera lenses with a finite aperture stop, and its length is infinitely small. Thick lens model [5] considers the length of a lens system. Matrix optics [6] derives the matrix form of refraction and reflection using the paraxial approximation. The matrix of a lens system is constructed by multiplying the matrices corresponding to lens elements. As to real-time rendering, image based lens model [7] and a single spherical lens element [8] are employed for rendering lens effects.

Full lens model. Kolb et al. [5] design a detailed lens model. Each element is modeled as a concrete geometrical body, therefore it incurs ray intersection tests. It is time-consuming to fire rays through a lens system with many elements. Afterwards, optical materials of lens elements are considered [10,11] to support spectral rendering of lens effects. Full lens model is also applicable to render advanced lens effects, like lens flares [9] and distortions [5].

Kolb et al. exploit an ideal exit pupil to guide camera rays through the lens system. Steinert et al. [10] propose a method to pre-compute valid pupils for individual pixels. Hanika and Dachsbacher [2] introduce an aperture based sampling method to pilot rays through the aperture, but the Newton iteration implementation adds extra cost for each ray. In this paper, we exploit virtual pupils based sampling. We pre-compute virtual pupils for line segments on the sensor.

Taylor polynomials. In the lens design area, polynomials have been applied to assist lens design [12], and to model optical aberrations [13]. They are, however, not designed for the rendering purpose.

Hullin et al. [1] express the refraction, reflection and propagation of rays as Taylor polynomials. The polynomial system of a lens system is constructed by sequentially substituting a polynomial system into the next one. Taylor expansion is conducted at the optical axis, and the precision of polynomials is not enough for off-axis points. 1-degree Taylor polynomials are applicable to render lens flares [14] at interactive frame rate.

Polynomial fitting. Hanika and Dachsbacher [2] introduce a fitting procedure to optimize the coefficients of Taylor polynomials. Recently, Schrade et al. [3] introduce the sphere/sphere parametrization at the front lens to support extreme wide-angle lenses. To speed up the evaluation, they utilize orthogonal matching pursuit [15] to select a few terms from all candidate terms. While it can find the optimal combination of terms, the search process costs nearly exponential time. In contrast, we propose an adaptive method to construct a sparse polynomial system, and it completes the work in polynomial time.

Feature selection. For polynomial regression, terms of a polynomial system can be viewed as features to describe the target system. Sequential forward search and backward search provide [16] preliminary solutions to select features from a fixed number of candidates. Adaptive basis function construction methods [17,18] build new features from existing ones, but it may produce a model with inadequate or excessive features if the search traps in a local optimum. Inspired by them, we adaptively construct a polynomial system to represent a lens system.

3 Polynomial regression framework

In rendering, camera lens model builds the connection between the light-carrying rays in a scene and incident rays on the camera sensor. The connection can be expressed as a function $\mathbf{\Phi} : \mathbb{V} \rightarrow \mathbb{V}$ which transforms an incident light field at the outer pupil to an outgoing light field at the sensor.

Polynomials have been widely used in optics to describe aberrations. In this paper, we use a polynomial system to approximate the function $\mathbf{\Phi}$. Given an incident ray R_i, the outgoing ray R_o can be calculated by a polynomial regressor $R_o = \mathbf{\Phi}(R_i)$. We use a 5D vector (x, y, u, v, λ) to represent R_i and another 5D vector $(\bar{x}, \bar{y}, \bar{u}, \bar{v}, \eta)$ for R_o (Fig. 1).

A polynomial system consists of five polynomials, each of which is a linear combination of multiple terms. The polynomial of the i-th output variable is

$$\mathbf{\Phi}_i = \sum_{a=0}^{\infty}\sum_{b=0}^{\infty}\sum_{c=0}^{\infty}\sum_{d=0}^{\infty}\sum_{e=0}^{\infty} \beta \cdot x^a y^b u^c v^d \lambda^e, \tag{1}$$

where β is the coefficient of a term and a, b, c, d, e are exponents. The degree of a term is defined as the sum of its exponents. The degree of a polynomial is defined as the maximum degree of terms.

We construct a polynomial system in a data-driven mode. We firstly build all terms, and then decide the coefficients of terms by fitting them to reference rays data. As to the change of focus distance, we move the sensor towards the rear lens to focus at a farther distance in the scene, and vice versa.

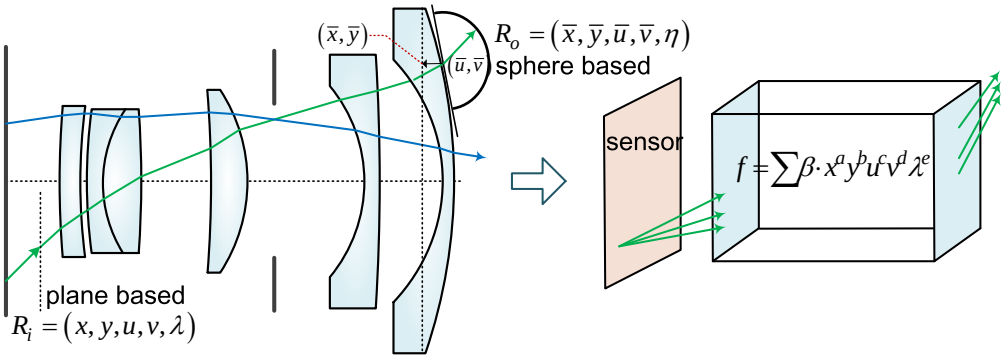


Fig. 1 Schematic of a fisheye lens (Rolf Muller 16mm, USP 4647161 [19])(*left*). A ray at the sensor with wavelength λ is expressed in plane based parametrization (x, y, u, v, λ). It is transformed to an outgoing ray on the outer pupil. The outgoing ray is denoted in a sphere based parametrization $(\bar{x}, \bar{y}, \bar{u}, \bar{v}, \eta)$ [3]. η is the Fresnel transmittance. The entire lens system can be abstracted as a polynomial system (*right*).

4 Adaptive sparse polynomial regression

Polynomials with high-degree terms can generally fit a complex non-linear target function. Given an allowed maximum degree d_{max} and p independent variables, the total number of possible terms is $\prod_{i=1}^{p}(1+d_{max}/i)$. Large value of d_{max} thus results in a combinatorial explosion of term count.

Taking into account the storage and evaluation of polynomials, it is reasonable to select a small number of important terms from optional terms. However, as the degree increases, it also becomes a problem to find the best combination of terms from a large number of candidate terms. Besides, it can be hard to decide an appropriate d_{max}.

In this paper, we construct terms of a polynomial system in a bottom-up approach. The construction process starts from the most basic term. It then adaptively constructs new terms using heuristic search. It does not require any preset degree threshold, and it is able to generate polynomials of arbitrary complexity.

The collection of terms of a polynomial can be viewed as a state. Constructing a polynomial is thus equivalent to search a state $\mathbf{\Phi}$ in the state space. We move from the current state to a new one by adding a term, deleting terms or modifying the degree of an existing term. The search is implemented in a greedy mode to minimize the sum of squared errors

$$\mathbf{\Phi} = \arg\min_{\mathbf{\Phi}} \sum_{(r_i,r_o)\in\mathbb{S}} \|r_o - \mathbf{\Phi}(r_i)\|^2, \tag{2}$$

where (r_i, r_o) is a reference ray sample in set $\mathbb{S}$. A state transition is performed only if the found new state leads to lower errors. After a state transition, we update the current accuracy achieved by the new state. The construction process stops if it cannot find a better state or the current accuracy satisfies the preset threshold.

4.1 Initial state

A term of the polynomial can be written as a product of the power of input variables: $\beta_i \cdot x^{a_{i1}} y^{a_{i2}} u^{a_{i3}} v^{a_{i4}} \lambda^{a_{i5}}$. Here, i is from 1 to infinity, $a_{ij} (j = 1 \cdots 5)$ is a non-negative integer exponent of the j-th variable. In our method, the initial state is the polynomial containing only the 0-degree term, which is also called the intercept term. It is used to construct high-degree terms via increasing exponents.

A polynomial with K terms can be recorded as a $K \times 6$ matrix (e.g. Eq. 3). The first column stores the coefficients of terms. The other five columns record exponents of each input variable. Thus, the initial state is a matrix with merely the first row of Eq. 3.

$$A = \begin{bmatrix} \beta_1 & 0 & 0 & 0 & 0 & 0 \\ \beta_2 & 1 & 0 & 1 & 2 & 0 \\ \cdots & & \cdots & & & \cdots \\ \beta_K & a_{K1} & a_{K2} & a_{K3} & a_{K4} & a_{K5} \end{bmatrix}. \tag{3}$$

4.2 Heuristic search operators

We define two types of heuristic search operators. Forward operators implement complication operations via adding new terms to the current state. Backward operators carry out simplification operations by deleting existing terms or decreasing the degree of a term. The change of degree is controlled by a step size parameter t. Figure 2 shows a part of the state transitions.

Forward operators. Forward operators will add new terms to the current state. It is necessary to ensure that a new term is not a duplicate of existing terms. We design two forward operators.

(F1) Adding a copy of an existing term and increasing one of its exponents by t.

(F2) Adding a new term with only one of its exponents being set to t and the others 0. The operator constructs

univariate t-degree terms. It is implemented only if the corresponding term is not included in the state.

Backward operators. Similarly, we define two backward operators to simplify the current state. The simplification is conducted via reducing unnecessarily high degree of terms or deleting unnecessary terms.
(B1) Selecting one of the existing terms, and decreasing one of its exponents by t. The remaining exponents should be non-negative, otherwise the decrease is not implemented.
(B2) Deleting one of the existing terms if the current state contains more than two terms.

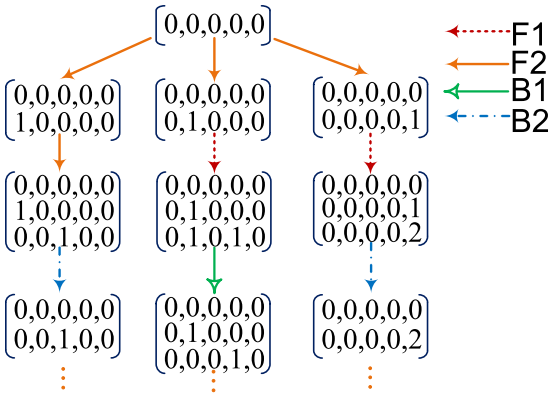


Fig. 2 State transitions in the state space. A state is expressed in a matrix form and coefficients are omitted for brevity.

4.3 Model fitting and evaluation

Since a polynomial is a linear combination of terms, we use the least-squares method to fit a polynomial model to training data. We adopt the Mean Squared Error (MSE) to measure the accuracy of a fitted polynomial model. After performing an operator, a set of candidate polynomials is constructed. Least-squares method is applied to obtain coefficients of polynomials and evaluate fitting errors. The polynomial with the minimum error is picked as the next state.

The construction process stops if errors of the polynomial model drop below preset thresholds. We use a threshold 10^{-7} for the output variables $\bar{x}$, $\bar{y}$ and η, whereas we use 10^{-10} for $\bar{u}$ and $\bar{v}$, because their orders of magnitude is smaller than other variables.

4.4 Replacement and recursion

Each term can be viewed as a feature to describe the camera lens. Polynomial models with a few terms can hardly fit complex nonlinear functions, whereas polynomial models with a large number of terms are prone to over-fitting if training data is not sufficient. In addition, polynomial models which contain too many terms also incur high computational cost. Many applications of machine learning limit the number of features to 40 [20]. We empirically find that 40 terms work well to approximate a camera lens.

The polynomial construction process is implemented in a greedy mode, and it may find more than 40 terms. If the number of terms exceeds 40, we perform a replacement operation, which replaces the least significant term in the current model with a newly found term.

Initially, the step size t is set to 1. The search in the state space may trap at a local optimum. We check the case and perform a recursion operation. If the construction process cannot find a better state, we increase t by 1. After t exceeds three, it empirically does not lead to better models, we stop the recursion if t exceeds three.

4.5 Algorithm and its complexity

Our method is listed in Algorithm 1. Its input is the initial state $\{\hat{t}\}$, the training set $\mathbb{S}$, the maximum number of terms Γ, and the recursion depth L.

In this section, we compare the computational complexity of the SHDP [3] and our method. SHDP assumes that the best model can be found in a predefined set $\mathbb{T}$ of terms within a degree p. For a polynomial with d independent variables, $\mathbb{T}$ contains $M = \binom{p+d}{p}$ terms. Large value of p results in combinatorial explosion of the term count.

SHDP assumes that the best model contains K terms. Initially, it searches the optimum combination of K terms from $\mathbb{T}$, and the number of states to be evaluated is $\binom{M}{K}$. Then, SHDP iteratively selects the $(K+1)$-th term from the remaining $M-K$ candidates. The number of evaluated states is $(M-K)\cdot F$, where F is the number of iterations. Afterwards, each existing term is replaced with the newly found one. The number of state to be evaluated is $(M-K)\cdot K\cdot F$. The approximate total number of evaluated states is

$$\binom{M}{K} + \binom{M-K}{1}\cdot(K+1)\cdot F \quad . \tag{4}$$

In our method, the branch factor of a state depends on both the number of variables d and the number of existing terms φ. For the operator (F2), the number of candidate states is d. For the operator (F1) and (B1), the count of candidate states is $\varphi\cdot d$. As to the operator (B2), it is φ. Therefore, the upper bound of candidate states is of the order $O(d+2d\varphi+\varphi) = O(d\varphi)$. Our

method adds one term to the model at a time, and the number of states to be evaluated is of the order

$$O(\sum_{i=1}^{K} di) = O(d\sum_{i=1}^{K} i) = O(d \cdot K(K+1)/2). \quad (5)$$

In the replacement step, the number of evaluated states is $dK \cdot E$, where E is the number of iterations. Each existing term is replaced with a newly found candidate, thus the number of state to be evaluated is $dK^2 \cdot E$. To sum up, the total number of evaluated states is of the order

$$O\left((\tfrac{1}{2}+E)dK^2 + (\tfrac{1}{2}+E)dK\right) = O\left((\tfrac{1}{2}+E)dK^2\right). \quad (6)$$

For instance, we substitute $d = 5, p = 11, K = 40$ into Eq. 4 and Eq. 6. The number of evaluated states of SHDP is $\binom{4368}{40} + 177448F$, whereas the result of our method is $4100 + 8200E$. Even though $E = F$, our method evaluates less candidate states. Thus, our method is more efficient in the construction process.

Algorithm 1 Adaptive sparse polynomial construction

Input: $\hat{t}$, $\mathbb{S}$, $\Gamma \leftarrow 40$, $L \leftarrow 3$, $depth \leftarrow 1$, $stuck \leftarrow false$
Output: A sparse polynomial system $\mathbb{T}$

$\mathbb{T} \leftarrow \{\hat{t}\}$
$bestError \leftarrow$ Evaluate($\mathbb{T}$);
while *true* **do**
  $T \leftarrow$FindBestNewTerm(F1, F2, $\mathbb{T}$, $depth$)
  if Evaluate($\mathbb{T} \bigcup \{T\}$)$<$ $bestError$ **then**
    if Size($\mathbb{T}$) $<$ Γ **then**
      $\mathbb{T} \leftarrow \mathbb{T} \bigcup \{T\}$
    else
      $(err, x) \leftarrow$FindBestReplace($(\mathbb{T} \backslash \mathbb{T}\{x\}) \bigcup \{T\}$)
      if $err >$ $bestError$ **then**
        $stuck \leftarrow true$;
      else
        $\mathbb{T}\{x\} \leftarrow T$; $bestError \leftarrow err$;
      if $bestError <$ eps **then**
        break;
  else
    break;
  $updated \leftarrow false$
  while *true* **do**
    $\mathbb{T}' \leftarrow$ Simplify(B1, B2, $\mathbb{T}$, $depth$)
    if Evaluate($\mathbb{T}'$) $<$ $bestError$ **then**
      $\mathbb{T} \leftarrow \mathbb{T}'$; $updated \leftarrow true$;
    else
      break;
  end while
  if Size($\mathbb{T}$)$==$ Γ && !$updated$ && $stuck$ **then**
    if $depth < L$ **then**
      $depth++$; $stuck \leftarrow false$
    else
      break;
end while
return $\mathbb{T}$

5 Light field partition

Aberrations are nonlinear functions of the aperture size and off-axis distance. The errors of Taylor polynomials also rise with the increase of off-axis distances [1]. Accounting for the facts, we propose to partition the incident light field at the sensor along the radial direction, and separately construct a polynomial system for each subregion.

The radius R of the paraxial region is set to $\alpha D/2$, where D is the diagonal length of sensor and α is in $[0.2, 0.3]$. In the paraxial region, the imaging properties are close to an ideal optical system, and it allows to use a low-degree polynomial system. Low-degree polynomials bring additional benefits that they are less costly to compute. In the off-axis region, it utilizes high-degree polynomials to better fit non-linear aberrations. To alleviate the discontinuity caused by the hard boundary, reference ray samples of the paraxial region are collected in a slightly larger circle with radius $(R+\epsilon)$ (Fig. 3). Similarly, ray samples of the off-axis region are collected outside of a slightly smaller circle with radius $(R-\epsilon)$. ϵ is a small positive number (e.g. 0.15).

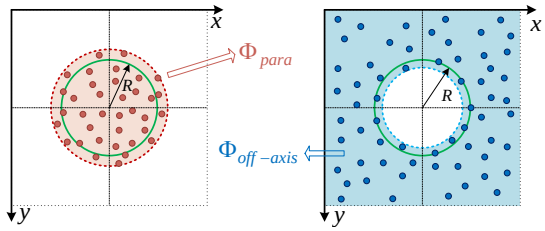


Fig. 3 Ray samples of the paraxial region are collected in the dashed red circle (*left*), and ray samples of the off-axis region are collected outside the dashed blue circle (*right*).

6 Line pupil–based rendering

Incident rays to a camera lens can be blocked by interior stops or the lens barrel. The reduction of survival rays causes the drop of rendering efficiency. We exploit line pupil–based sampling to pilot rays through lens elements. Camera lenses are generally designed to be rotationally symmetric around the optical axis. Similar to the method [21], we compute line pupils for line segments along the radial direction outwards. Line pupils change with the positions on the sensor. In the paraxial region, we divide the radial line into N segments with length $\frac{\alpha D}{2N}$. In the off-axis region, we use a shorter length $\frac{\alpha D}{4N}$ to better capture the change of valid pupils.

For a line segment a, we fire a ray from a point on a to a point P on the rear plane (Fig. 4). If the ray can pass the lens system, P is included in the range of the

line pupil. Finally, the bounding box of points on the rear plane decides the line pupil. After stopping down the aperture, line pupils should be updated accordingly while keeping the polynomial system unchanged. Front pupils can be pre-computed in a similar approach. A starting plane is firstly selected before the front lens.

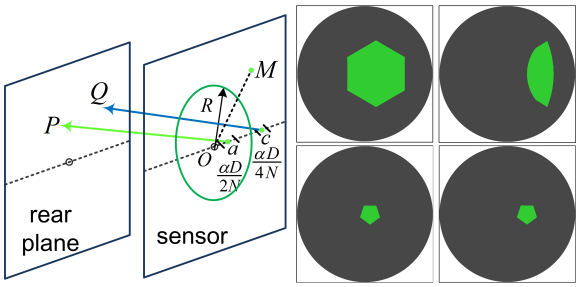


Fig. 4 The configuration of calculating line pupils. The right side are line pupils of line segments a and c. (*top*) A gauss lens (F/4.0) with hexagon aperture, (*bottom*) a fisheye lens (F/12.0) with pentagon aperture.

At runtime, a starting point M of a ray is sampled at the sensor. Depending on its distance to the optical axis, it is located in a line segment. Then we rotate its line pupil with the angle between the horizontal radial line and the radial line of M. Afterwards, the ending point of the ray is sampled in the rotated line pupil.

While the method is effective to guide rays through a camera lens, there are also invalid rays because the line pupil is a loose bounding box. To reduce invalid rays, we add the aperture constraints of the front lens and the rear lens. Rays that cannot pass the two lenses are clipped.

6.1 Monte Carlo estimator

In most cases, a camera lens is relatively small compared to the scene size. It is hard for an arbitrary incoming ray to hit the front lens. In addition, the incident ray will go through specular bounces, forming a "SDS" path, which is hard to sample. Therefore, we do not use light tracing from the light to the sensor. Only a unidirectional polynomial system is constructed.

In Monte Carlo ray tracing, the radiance of a pixel can be described as an integral of a measurement contribution function $f = L(s_1, -\omega_0)\bar{G}(s_1 \leftrightarrow s_0)W_e(s_0)$, where L is the incident radiance to s_1, $\bar{G}(s_1 \leftrightarrow s_0)$ is an extended factor and W_e denotes the visual importance. Path vertices $s_0, s_1, \cdots, l_1, l_0$ are shown in Fig. 5.

The camera path starts from point s_0 sampled with a probability density function $p(s_0)$, and a point s_0' is sampled with $p(s_0')$. s_0' is an artificial point for assisting

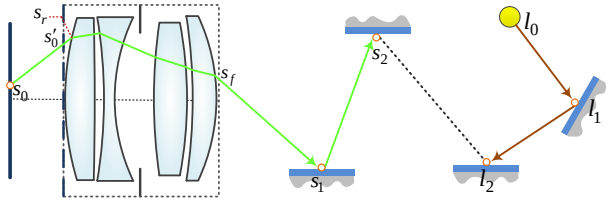


Fig. 5 Camera subpath and light subpath in the context of rendering with complex camera lenses. s_0 is on sensor and s_0' is on a line pupil. $\overline{s_0 s_1}$ is treated as a virtual edge of the camera subpath.

the path construction. The estimator is defined as

$$\frac{1}{N}\sum \frac{L(s_1, -\omega_0)\bar{G}(s_1 \leftrightarrow s_0)W_e(s_0)}{p(s_0)p(s_1)}. \tag{7}$$

Here, we transform the area measure of s_1 to the area measure of s_0': $p(s_1) = p(s_0') \cdot |J_{all}| \cdot \left|\frac{\partial s_1}{\partial s_0'}\right|$, where $|J_{all}|$ is a combined Jacobian for density transformation. $\bar{G}(s_1 \leftrightarrow s_0)$ is defined as:

$$\bar{G}(s_1 \leftrightarrow s_0) = G(s_1 \leftrightarrow s_0) \cdot P_a(s_1 \leftrightarrow s_0) \cdot T_r, \tag{8}$$

Here, G is a generalized geometric term [22]. P_a denotes a passage function, it is 1 if the ray can pass the lens system and 0 otherwise. T_r is the Fresnel transmittance. Finally, the estimator can be simplified as:

$$\frac{1}{N}\sum \frac{L \cdot W_e \cdot P_a \cdot T_r}{p(s_0)p(s_0')} \cdot \frac{\cos\theta_{s_0}\cos^2\theta_{s_1}\cos\theta_{s_f}}{\cos^2\theta_{s_0'} \cdot \|s_f - s_1\|^2}. \tag{9}$$

The detailed derivation is placed in the supplemental document.

In bidirectional path tracing, a path with k edges can be sampled with $k+2$ techniques. Since we omit the light tracing, the remaining number of techniques is $k+1$. Unbiased result can be obtained as long as samples are correctly weighted. Each possible path of length k should be reweighted and the sum of all weights is 1. We use an equal weight $\frac{1}{k+1}$.

7 Results

7.1 Analysis

Our experiments are conducted on a workstation with 2.4 GHz Intel Xeon CPU. The number of terms is limited to 40. A fitted polynomial system is shown in the supplemental document.

Accuracy and efficiency comparisons. Table 1 shows the fitting statistics of three camera lenses from [3]. For fair comparisons, each method fits a camera lens as a single polynomial system. The last row shows our method with light field partition. The error thresholds for SHDP and our method are the same.

The training data set contains 3K samples. All models are tested with respect to a testing data set of 50K samples. Taylor polynomial models [1] are constructed in an analytic fashion, and its timings are not included. As seen, our method takes less time to construct a polynomial system. The accuracy of SHDP and OUR-single is similar. Note that, our method with light field partition slightly improves the general accuracy.

Table 1 Comparisons of the average sum of squared errors and fitting time. 'OUR-single' fits a camera lens as a single polynomial system, whereas 'OUR' fits two polynomial systems.

Camera	double-gauss	fisheye-ii	canon-anamorphic
Taylor (Deg3)	$4.74 \cdot 10^{-3}$	$3.39 \cdot 10^{-3}$	$2.30 \cdot 10^{-3}$
Taylor (Deg5)	$2.61 \cdot 10^{-4}$	$2.78 \cdot 10^{-3}$	$1.39 \cdot 10^{-3}$
Taylor (Deg7)	$1.21 \cdot 10^{-4}$	$1.98 \cdot 10^{-4}$	$8.79 \cdot 10^{-4}$
SHDP	$4.85 \cdot 10^{-6}$ 138.3h	$1.39 \cdot 10^{-4}$ 180.14h	$2.30 \cdot 10^{-3}$ 183.2h
OUR-single	$5.05 \cdot 10^{-6}$ 24.6h	$1.10 \cdot 10^{-4}$ 26.8h	$2.50 \cdot 10^{-3}$ 30.1h
OUR	$3.05 \cdot 10^{-6}$ 51.7h	$5.79 \cdot 10^{-5}$ 56.3h	$6.72 \cdot 10^{-4}$ 55.6h

The absolute errors for simulating the fisheye-ii lens are shown in Fig. 6. We build rays by connecting points on the horizontal axis of sensor to a fixed point on the vertical axis of rear pupil. The fixed point is at the 1/3 radius position. With light field partition, our method generally fits polynomials with high accuracy. Note that it gives lower errors in the paraxial region.

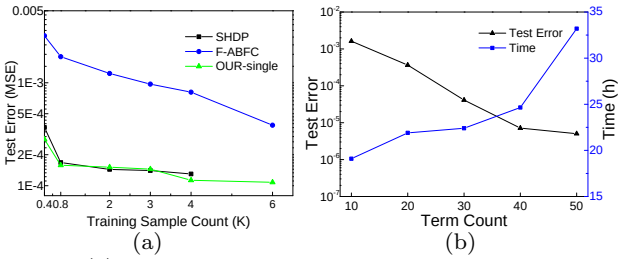
Number of samples and terms. Table 2 shows the fitting time as to the number of samples. The fisheye-ii lens is fitted here. All methods use the same error thresholds. Our method costs less time than others. The time cost of SHDP goes up quickly, because it involves costly Cholesky decomposition, whose time cost rises quickly as the size of matrix increases. Both F-ABFC [18] and our method tend to use high-degree terms if they are trained with a small number of samples. Nevertheless, over-fitting is likely to occur. With the increase of degree, its generalization ability drops if samples are not enough. As shown in Fig. 7a, the high-degree polynomials result in higher testing errors. Given more samples, our method can fit the samples with low-degree polynomials, which are desirable.

Figure 7b plots the test error and the fitting time as to different number of terms. The fitting time goes up quickly as the increase of term count. 30–40 terms can provide adequate imaging accuracy and cost moderate time.

Passage Rates. we compare the passage rates of rays as to three camera lenses (Fig. 8). Note that there are no stops in a polynomial system, we therefore add constraints of the front lens and the rear lens to SHDP. As shown, the rear pupil based method [5] gives low passage rates. The ideal exit pupil based method [5] improves the rates, but cannot achieve 100%, because it is calculated from the paraxial region. Our line pupil–based method further raises the rates. SHDP provides higher survival rates. Note that its Newton iteration may exit early without convergence, thus causing invalid rays.

Table 2 Comparisons of the execution time and the maximum degrees of polynomials (placed in parentheses). Note the case of 6000 samples of SHDP is not available.

Sample#	SHDP	F-ABFC	OUR-single
400	150.4h (11 per each)	32.5h (12,13,10,11,15)	3.05h (8,10,9,9,14)
2000	162.5h (11 per each)	67.5h (11,10,11,9,13)	19.5h (7,7,5,5,12)
3000	180.1h (11 per each)	88.5h (11,9,10,10,13)	26.8h (5,6,8,5,11)
4000	1028.3h (11 per each)	109.7h (9,8,9,10,11)	32.7h (5,6,5,5,11)
6000	>1000h (-)	143.5h (8,6,9,8,10)	46.3h (5,6,6,5,9)



(a) (b)

Fig. 7 (a) Testing errors of the polynomials fitted with different number of samples and (b) analysis of the test error and fitting time as to the number of terms.

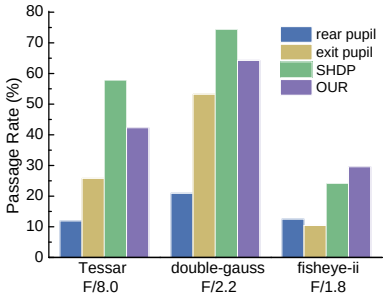


Fig. 8 Passage rate comparisons of 4 methods: full lens model with rear pupil sampling, full lens model with ideal exit pupil sampling, aperture based sampling [3] and our method.

7.2 Applications in rendering

Figure 9 are shot by a simple bi-convex lens. Primary space Metropolis light transport (PSMLT) [23] is adopted for rendering. It illustrates that our polynomial system can produce typical Seidel aberrations like

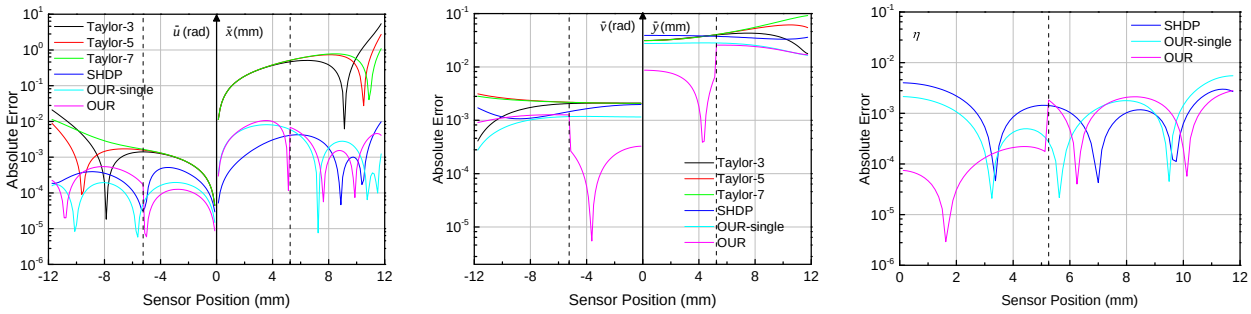


Fig. 6 Comparisons of absolute errors of Taylor polynomial models from degree 3 to degree 7, SHDP, our model fitted as a whole, and our model with partitions. (*left*) $\bar{x}$ and $\bar{u}$, (*middle*) $\bar{y}$ and $\bar{v}$ and (*right*) η. Note that Taylor polynomial models do not contain a transmittance dimension. The black dashed line shows the position of partition ($\alpha = 0.3, R = 5.25$).

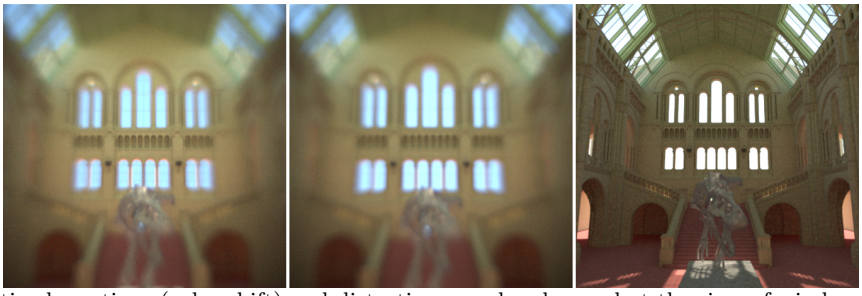


Fig. 9 (*left*) Chromatic aberrations (color shift) and distortions can be observed at the rims of windows. (*middle*) The sensor is moved backward and the interior frames of windows are blurred. (*right*) Only 1-degree terms are used and it is similar to the result of a thin lens model.

distortions and chromatic aberrations (see the rims of windows).

In Fig. 10, the same bi-convex lens is used for the chess scene. The focal distance is adjusted via moving the sensor plane. Bokeh effects are shown at the defocus region. The color rings of the bokeh patterns depict the chromatic aberrations. The shape of the circle of confusion is affected by the shape of aperture.

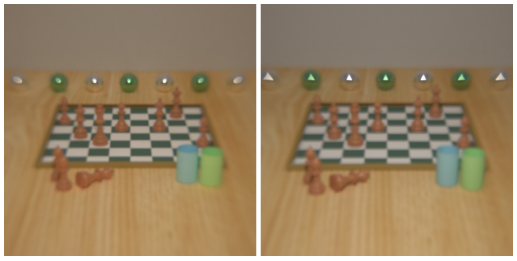


Fig. 10 Bokeh rendered by our model. (*left*) Focal distance is set to 1000 and the aperture is circular. (*right*) Focal distance is 4000 and the aperture is triangular.

Figure 11 compares the sponza scene photographed by a fisheye-ii lens. The rendering method is bidirectional path tracing (BPT) [24]. For the Taylor polynomial model, we apply our Monte Carlo estimator (Sect. 6.1) to it to support BPT. The Taylor polynomial model leads to an overall blur image. Note the blur is a kind of systematic noise caused by inefficient imaging precision, instead of defocus blur. SHDP is noisier because it spends more time on iterative adaptation of the direction of a ray at runtime. At the image periphery, our method provides clearer texture details. This is enabled by the strict clipping of invalid rays.

In Fig. 12, the sala scene is imaged with a double-gauss lens and the stochastic progressive photon mapping method [25] is used here. Taylor polynomial model results in different distortions at the upper edge of the image. There is noticeable color contouring (close-up views of Fig. 12a). Both SHDP and our method produce image profiles close to the reference. After the same rendering time, our method provides an image with better visual quality and lower numerical error.

8 Discussion

Sample count and degree limit. SHDP [3] suggests using 10 times more samples than the term count. For an 11-degree polynomial, SHDP considers 4368 candidate terms at each step. 10 times more samples lead to considerable fitting time. In contrast, our method considered at most 41 terms at each step. Therefore, 410 samples satisfy the requirements of sample count. Given more than 3000 samples, our method can build a polynomial system within 11 degree. Therefore, pro-

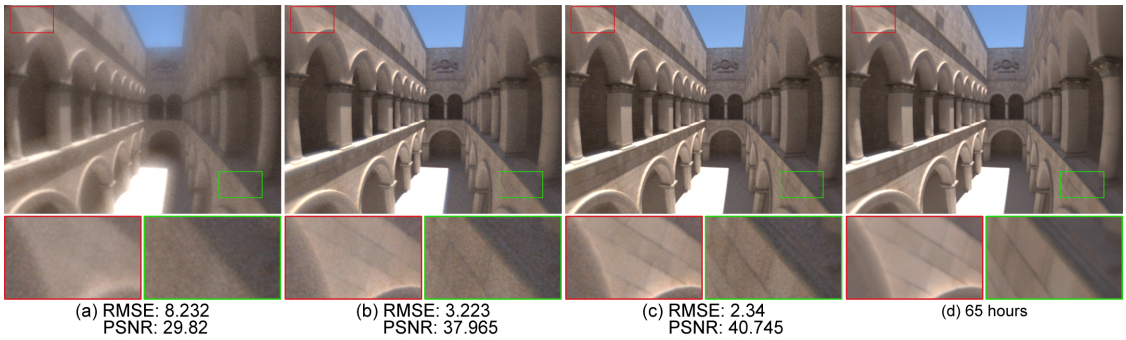


Fig. 11 Equal time comparison (2h) of (a) the Taylor polynomial model of 3-degree, (b) SHDP model and (c) our model. SHDP and our method are fitted to 4000 samples. The reference image (d) is rendered with a full lens model using 65 hours.

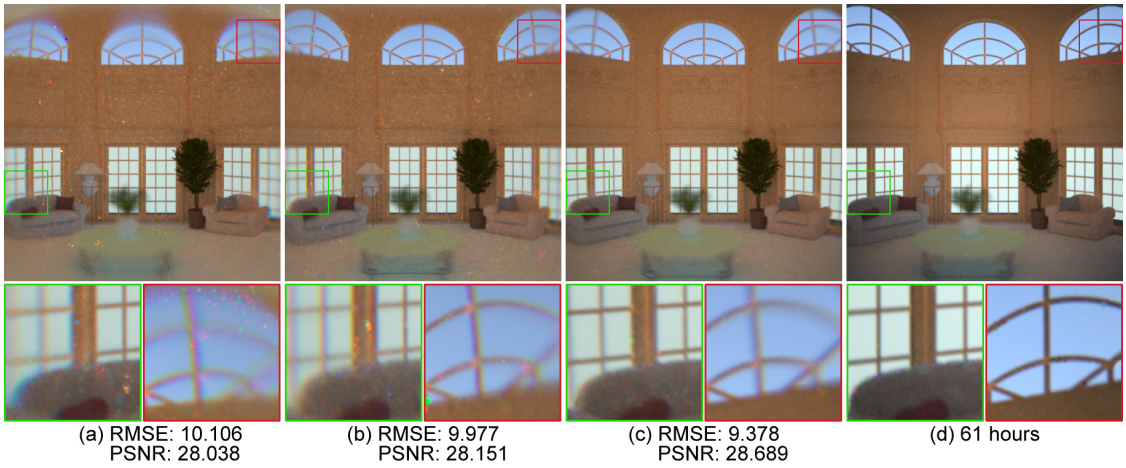


Fig. 12 Equal time comparison (3.5h) of the (a) Taylor polynomial model of 3-degree, (b) SHDP model and (c) our model. The reference image (d) is rendered with a full lens model using 61 hours.

vided with enough samples, our method does not limit the degree.

Metropolis light transport. PSMLT is efficient to sample difficult paths in Fig. 9. A path of PSMLT corresponds to a set of random samples. The lens samples are now used to sample points on the line pupil rather than on the aperture. After a mutation operation, the new sensor samples may correspond to a new line pupil. However, the subsequent mutated lens samples may not produce a point within the new line pupil and leads to invalid camera rays. Thus we use uniform random numbers for sampling on the line pupil.

Limitations and future work. Our method provides a polynomial model to approximate a complex camera lens. There are no constraints of lens stops in a polynomial system. Therefore, more camera rays are transported into the scene, causing a slightly brighter image.

Our method approximates a complex camera lens in the sense of geometric optics. Lens effects based on physical optics, like interference and diffraction are currently not supported. Polynomial models have been demonstrated to be applicable to render lens flares [1, 14]. Our model for general image formation cannot be directly applied to render lens flare. It would be interesting to construct specialized polynomials for lens flare rendering using our algorithm.

Besides, our method does not support zoom lens. The change of the zoom ratio leads to a new polynomial system. It can be constructed by replacing terms of the current polynomial model. This may save computation time than starting from scratch.

9 Conclusion

This paper has proposed a novel adaptive method to construct a polynomial system for approximating a complex camera lens. Our approach starts from low-degree terms and it then automatically builds necessary high-degree terms. Depending on the properties of aberrations, we partition the light field at the sensor along radial direction and handle subregions separately. Replacement operation is utilized to replace existing terms with more significant terms. Recursion operation is exploited to avoid trapping at a local optimum. Besides, we use line pupil–based method to generate camera rays to improve the survival rates of rays. Compared with state-of-the-art methods, our approach behaves more efficient at building a sparse polynomial

system, and it can achieve high imaging accuracy at the same time.

Acknowledgements This work was supported in part by the research grant (ref. 9140A21010115HT05003). The camera lens data is courtesy of Emanuel Schrade et al.

References

1. Hullin, M.B., Hanika, J., Heidrich, W.: Polynomial optics: A construction kit for efficient ray-tracing of lens systems. Comput. Graph. Forum **31**(4), 1375–1383 (2012)
2. Hanika, J., Dachsbacher, C.: Efficient Monte Carlo rendering with realistic lenses. Comput. Graph. Forum **33**(2), 323–332 (2014)
3. Schrade, E., Hanika, J., Dachsbacher, C.: Sparse high-degree polynomials for wide-angle lenses. Comput. Graph. Forum **35**(4), 89–97 (2016)
4. Potmesil, M., Chakravarty, I.: A lens and aperture camera model for synthetic image generation. ACM SIGGRAPH Computer Graphics **15**(3), 297–305 (1981)
5. Kolb, C., Mitchell, D., Hanrahan, P.: A realistic camera model for computer graphics. In: Proceedings of SIGGRAPH '95, SIGGRAPH '95, pp. 317–324. ACM, New York, NY, USA (1995)
6. Gauss, C.F.: Dioptrische untersuchungen. Dioptrische Untersuchungen, by Gauss, Carl Friedrich, 1841. **1** (1841)
7. Heidrich, W., Slusallek, P., Seidel, H.P.: An image-based model for realistic lens systems in interactive computer graphics. In: Graphics Interface, vol. 97, pp. 68–75 (1997)
8. Lee, S., Eisemann, E., Seidel, H.P.: Real-time lens blur effects and focus control. ACM Trans. Graph. **29**(4), 65:1–65:7 (2010)
9. Hullin, M., Eisemann, E., Seidel, H.P., Lee, S.: Physically-based real-time lens flare rendering. ACM Trans. Graph **30**(4), 108:1–108:10 (2011)
10. Steinert, B., Dammertz, H., Hanika, J., Lensch, H.P.: General spectral camera lens simulation. Comput. Graph. Forum **30**(6), 1643–1654 (2011)
11. Wu, J., Zheng, C., Hu, X., Xu, F.: Rendering realistic spectral bokeh due to lens stops and aberrations. Vis. Comput. **29**(1), 41–52 (2013)
12. Zheng, N., Hagen, N., Brady, D.J.: Analytic-domain lens design with proximate ray tracing. JOSA A **27**(8), 1791–1802 (2010)
13. Hopkins, G.W.: Proximate ray tracing and optical aberration coefficients. JOSA **66**(5), 405–410 (1976)
14. Lee, S., Eisemann, E.: Practical real-time lens-flare rendering. Comput. Graph. Forum **32**(4), 1–6 (2013)
15. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory **53**(12), 4655–4666 (2007)
16. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognit. Lett. **15**(11), 1119–1125 (1994)
17. Todorovski, L., Ljubič, P., Džeroski, S.: Inducing polynomial equations for regression. In: European Conference on Machine Learning, pp. 441–452. Springer (2004)
18. Jekabsons, G., Lavendels, J.: Polynomial regression modelling using adaptive construction of basis functions. In: Proceedings of IADIS International Conference, Applied Computing, pp. 269–276 (2008)
19. Smith, W.J.: Modern lens design, 2 edn. McGraw-Hill New York, Maidenhead, Berkshire, England (2005)
20. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**(Mar), 1157–1182 (2003)
21. Pharr, M., Jakob, W., Humphreys, G.: Physically based rendering: From theory to implementation, 3 edn. Morgan Kaufmann (2016)
22. Jakob, W., Marschner, S.: Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. ACM Trans. Graph. **31**(4), 58:1–58:13 (2012)
23. Kelemen, C., Szirmay-Kalos, L., Antal, G., Csonka, F.: A simple and robust mutation strategy for the metropolis light transport algorithm. Comput. Graph. Forum **21**(3), 531–540 (2002)
24. Lafortune, E.P., Willems, Y.D.: Bi-directional path tracing. In: Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93), pp. 145–153 (1993)
25. Hachisuka, T., Jensen, H.W.: Stochastic progressive photon mapping. ACM Trans. Graph. **28**(5), 141:1–141:8 (2009)