On the Maximum Radius of Polynomial Lens Distortion

Matthew J. Leotta
Kitware, Inc.
matt.leotta@kitware.com

David Russell*
Carnegie Mellon University
davidrus@andrew.cmu.edu

Andrew Matrai
Kitware, Inc.
andrew.matrai@kitware.com

Abstract

Polynomial radial lens distortion models are widely used in image processing and computer vision applications to compensate for when straight lines in the world appear curved in an image. While polynomial models are used pervasively in software ranging from PhotoShop to OpenCV to Blender, they have an often overlooked behavior: polynomial models can fold back onto themselves. This property often goes unnoticed when simply warping to undistort an image. However, in applications such as augmented reality where 3D scene geometry is projected and distorted to overlay an image, this folding can result in a surprising behavior. Points well outside the field of view can project into the middle of the image. The domain of a radial distortion model is only valid up to some (possibly infinite) maximum radius where this folding occurs. This paper derives the closed form expression for the maximum valid radius and demonstrates how this value can be used to filter invalid projections or validate the range of an estimated lens model. Experiments on the popular Lensfun database demonstrate that this folding problem exists on 30% of lens models used in the wild.

1. Introduction

The geometric computer vision literature is largely built around the pinhole model of a camera. Projective geometry concepts like projection matrices, homographies, fundamental matrices, and so on are built upon the mathematical image formation model that ensures that straight lines in the world coordinates project into straight lines in the image plane. Yet, real world lenses often have distortion. These distortion effects must be modeled in order to apply the power of the pinhole model in many computer vision applications without significant measurements errors [8].

The most common approaches for correcting distortion are radial distortion models. Radial distortion assumes that the distortion is radially symmetric about a center of distortion—often the center of the image. Points along the radial lines are scaled by a non-linear function of the radius. While there are many variants of this distortion model that have been explored, in practice a simple polynomial function is most often used. The Brown–Conrady [4] model is commonly cited in the literature when referring to this polynomial model. The Brown–Conrady model uses even degree polynomial terms, but other variants exist using both odd and even terms. The polynomial coefficients are typically estimated from image data in one of three ways: optimizing the projection of known 3D points from calibration target (like a checkerboard) [31, 3, 33], bundle adjustment of points matched across multiple views [13, 2, 15, 22, 25],

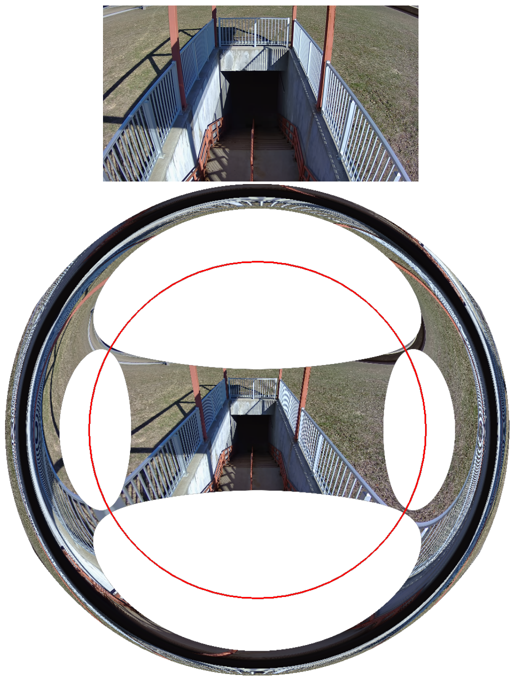


Figure 1: Example of undistortion. Beyond the red circle (maximum radius of distortion) the distortion folds back on itself. Points outside the maximum radius may incorrectly project into the valid image bounds causing problems in practical vision applications.

*Contributed to this work while employed at Kitware, Inc.

or by minimizing the curvature of lines that are known to be straight [10, 5, 26, 1, 27, 30, 32]. While various alternative models have been proposed in the literature, the majority of production software applications use some variant of a polynomial model to account or correct for radial distortion.

Most computer vision courses and textbooks [16, 28, 14] teach polynomial lens distortion models but typically as a side topic and not in much depth. Students of computer vision commonly estimate and apply these models only to warp an image to remove the distortion effects. As a result, not much attention is given to how these models behave outside the bounds of the image. Polynomial lens distortion models have an often overlooked property that may lead to surprising results when using radial lens distortion in practical applications. **Radial lens distortion models may fold back on themselves** allowing points in front of the image plane but far outside the field of view to project back into the image. Practically, applications in which 3D scene content is projected into an image and warped into the original distorted image space, such as augmented reality, are likely to encounter these invalid projections. In extreme cases, where the model is poorly fit to the periphery of the image, these folding artifact materialize in corners of images when warping to undistorted coordinates. These folding artifacts are commonly encountered in practice and reported in support forums [20, 23, 24] whose discussions lack theoretical explanation of the problem at hand. This paper aims to provide that missing explanation along with an analysis of when it occurs and an algorithm to detect it. Figures 1 and 2 illustrate this problem in different ways.

Figure 1 shows an example of what can happen when warping an image to remove distortion if the range of the output image is expanded. The red circle shows the maximum radius for which the distortion function is bijective. Outside of this circle pixels may be erroneously sampled from the middle of the image. In fact, there is a second larger radius for which all points on this circle map to the center of distortion in the middle of the image. If the maximum radius is far outside the image bounds these artifacts will go unnoticed during warping. However, if this radius falls within the image the artifacts cannot be avoided.

Figure 2 illustrates the problem geometrically and shows how a single straight line is distorted to curve back into the image. The camera effectively has a secondary invalid field of view (FOV) lying outside of the true FOV. Any points in the invalid FOV will incorrectly project into the image after distortion. This is analogous to the pinhole camera property that mathematically projects points both in front and behind the camera into the image. Most students of projective geometry will know to filter points that lie behind the camera, but fewer are aware of the need to filter points outside the maximum valid radius of distortion.

The contributions of this paper are three-fold. First,

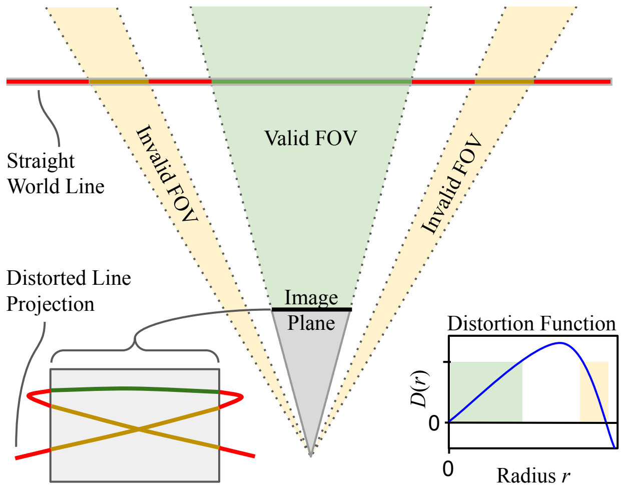


Figure 2: Top view cross section of a camera frustum showing the projection of a straight line in the world into a curved line in the distorted image. The green portion of the line falls within the valid field of view (FOV) and maps correctly. The red portion of the line maps outside the image, as expected. The yellow part of the line should project outside the image but incorrectly folds back into the image due to the distortion function. Any points in the yellow invalid FOV regions will incorrectly distort into the image.

this paper provides a public service announcement to raise awareness of the types of invalid image projections that can occur when using polynomial radial distortion models in practical applications. Second, the paper describes the conditions under which a distortion model will fold back on itself and derives the closed-form expression for the maximum valid radius for polynomials up to third degree. The maximum valid radius marks a circular domain such that the distortion function is bijective in that domain. The solution to the third degree case is nontrivial to derive as it requires roots of a cubic polynomial. A simple algorithm is given to compute the maximum radius of distortion to use as a filter when projecting points or to check that an estimated model is valid for the entire image domain. Third, experiments with the Lensfun [9] database of distortion models, which is used in many production software systems, demonstrate how frequently the folding issue occurs in the wild. Open source code for work in this paper is available online[1].

2. Related Work

Radial lens distortion models have been studied for a very long time. One of the most commonly cited polynomial models, the Brown–Conrady model, was developed over a half century ago by Brown [4] building on initial work by Conrady [7] another half century before that.

[1]`https://github.com/Kitware/max-lens-radius`

While other types of lens distortion models exist in the literature, polynomial models are simple and commonly used in practice. This section focuses on polynomial models, which are relevant to this paper, but also briefly surveys other lens distortion models.

2.1. Radial Polynomial Models

Radial distortion models compensate for lens distortion by scaling the image radially about a center of distortion as a function, $F(r)$, of the radius, r. Distortion is most commonly applied in normalized image coordinates to an undistorted point, (x_u, y_u), where the center of the distortion is at the origin and the radius is therefore $r_u = \sqrt{x_u^2 + y_u^2}$. The distorted point (x_d, y_d) is computed as

$$\begin{aligned} x_d &= x_u F(r_u), \\ y_d &= y_u F(r_u). \end{aligned} \tag{1}$$

Likewise one can define the distorted radius as

$$r_d = D(r_u) = r_u F(r_u). \tag{2}$$

There are several polynomial variations of $F(r)$. The basic third degree polynomial has the form

$$F_P(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3, \tag{3}$$

with model coefficients $\{k_1, k_2, k_3\}$. Equation (3) can be viewed as a third order Taylor series expansion of the ideal distortion function. While additional higher order terms can be added as needed, the literature generally agrees that a third order polynomial is sufficient for most real lenses [16, 28, 14]. Often only one or two coefficients are used in cases with less severe distortion.

The Brown–Conrady model [4] is a variant of the form

$$F_B(r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6. \tag{4}$$

Note that this sixth degree polynomial still has only three coefficients since only the even powers of r appear. Using only even powers avoids the need for a square root operation as one can compute Equation (4) directly from r^2. Conveniently, one can express the Brown–Conrady model in terms of Equation (3) using the change of variables $\hat{r} = r^2$ such that $F_B(r) = F_P(\hat{r})$.

The model proposed by Brown also includes additional non-radial terms called the tangential components. Tangential distortion compensates for manufacturing defects where the lens is not parallel to the sensor plane. Since manufacturing quality of cameras is high, tangential distortion is rarely used in practice and is ignored in the remainder of this paper.

2.2. Other Lens Distortion Models

Other distortion models include the rational models of Hartley and Saxena [17] or Claus and Fitzgibbon [6], the Field of View model of Devernay and Faugeras [10], and the division model of Fitzgibbon [13]. A more detailed review and comparison of alternate radial distortion models can be found in [29]. While these models offer some clear advantages over the polynomial model, especially in very wide angle and fish-eye lenses, the simple polynomial model remains the most widely used in computer vision applications. The pervasiveness of polynomial models is likely due to its simplicity and the availability of open source tools for fitting these models, such as the OpenCV [21] implementation of Zhang's method [33].

The purpose of this paper is not to propose a new mathematical model for lens distortion to address the shortcomings of the popular polynomial model. Rather, the purpose of this paper is to provide a deeper understanding of those shortcomings so that applications committed to the use of polynomial models may continue to use them while avoiding the pitfalls of doing so. The remainder of this paper focuses on polynomial distortion.

3. Computing the Maximum Valid Radius

Any radial distortion function, $D(r)$, will either grow monotonically to infinity or will reach a local maximum after which slope of $D(r)$ becomes negative. If the slope is negative then the distortion folds back on itself and $D(r)$ is no longer a bijection (not invertible). Thus, the maximum valid radius, $r_{\max}$, occurs at the smallest positive local maximum of $D(r)$. If $D(r)$ is monotonic then $r_{\max} = \infty$. To solve for $r_{\max}$ we need to find the smallest positive root of $D'(r)$, the derivative of $D(r)$ with respect to r. Note that $D'(r)$ can have multiple positive roots, but we are only concerned with the smallest root to find the maximum domain of $r \in (0, r_{\max})$ for which $D(r)$ is monotonic.

The derivatives for the basic and Brown–Conrady distorted radii set equal to zero are

$$D'_P(r) = 1 + 2k_1 r + 3k_2 r^2 + 4k_3 r^3 = 0 \tag{5}$$

$$\text{and} \quad D'_B(r) = 1 + 3k_1 r^2 + 5k_2 r^4 + 7k_3 r^6 = 0, \tag{6}$$

respectively. Note that $F(r)$ is a multiplicative factor applied to the undistorted radius as described by Equation (2). Therefore, $D'_P(r)$ (resp. $D'_B(r)$) is a polynomial of the same degree as $F_P(r)$ (resp. $F_B(r)$), just with different coefficients. Both Equations (5) and (6) can be reduced to the more general form

$$1 + aw + bw^2 + cw^3 = 0. \tag{7}$$

In both cases, the constants ([2, 3, 4] or [3, 5, 7]) can be folded into coefficients a, b, and c. Furthermore, for Equation (6), the substitution $w = r^2$ reduces the problem to a third degree polynomial. The final solution for Equation (6) is then the square root of the solution to Equation (7).

3.1. First and Second Degree Solutions

For a first degree polynomial, where $b = c = 0$, the solution to Equation (7) is trivially $w = \frac{1}{-a}$. The second degree polynomial, with only $c = 0$, is also easily solved by the quadratic equation where $w = \frac{-a \pm \sqrt{a^2-4b}}{2b}$. Note that the second degree solution is ill-defined when $b = 0$. Using a lesser known form of the quadratic equation from Muller's method [19] there is a solution that covers both the linear and quadratic cases consistently:

$$w = \frac{2}{-a + \sqrt{a^2 - 4b}}. \tag{8}$$

Only the positive square root is needed here since we seek smallest positive real root. If the the discriminant, $a^2 - 4b$, is negative or if the denominator of Equation (8) is less than or equal to zero, then there are no positive roots. In this case, $r_{\max} = \infty$.

3.2. Third Degree Solution

The cubic formula for the roots of Equation (7) in the general case has been known for centuries [11], but it is considerably more complicated to express in closed form in terms of a, b, and c. We can, however, write the solution more compactly by defining several intermediate variables:

$$\beta = 2b^3 - 9abc + 27c^2 \tag{9}$$

$$\gamma = b^2 - 3ac \tag{10}$$

$$\Delta = \beta^2 - 4\gamma^3 \tag{11}$$

$$U = \lhd \sqrt[3]{\frac{\beta + \sqrt{\Delta}}{2}} \tag{12}$$

$$\lhd \in \left\{1, \frac{-1+\sqrt{3}i}{2}, \frac{-1-\sqrt{3}i}{2}\right\} \tag{13}$$

Given the above definitions, the solution takes the form

$$w = \frac{-1}{3c}\left(\frac{\gamma}{U} + U + b\right) \tag{14}$$

with the three solutions determined by the three possible values of $\lhd$ in Equation (13). Here the discriminant is Δ. If $\Delta \geq 0$ then the square root in Equation (12) is real valued and there is only one real valued cubic root corresponding to the $\lhd = 1$ case.

If $\Delta < 0$ then there are three real valued solutions, but the computation of those real values involves the cube root of a complex number, and it is not at all obvious how to evaluate it. In this case, it is easier to write Equation (12) in polar coordinates. For a complex number of the form $z = x + iy$, the polar coordinates are $z = re^{i\theta}$ with radius $r = \sqrt{x^2 + y^2}$ and angle $\theta = \operatorname{atan2}(y, x)$. Using these identities

$$U = \lhd \sqrt[3]{\frac{\beta + i\sqrt{-\Delta}}{2}} \tag{15}$$

$$= \lhd \sqrt[3]{\frac{1}{2}\sqrt{\beta^2 + \sqrt{-\beta^2 + 4\gamma^3}^2}\, e^{i\theta}} \tag{16}$$

$$= \lhd \sqrt[3]{\frac{1}{2}\sqrt{4\gamma^3}\, e^{i\theta}} \tag{17}$$

$$= \lhd \sqrt{\gamma} e^{i\theta/3} \tag{18}$$

where $\theta = \operatorname{atan2}(\sqrt{-\Delta}, \beta)$. Note that $\Delta < 0$ also implies $\gamma > 0$, so the square root above is real valued. In polar coordinates $\lhd = \{1, e^{i2\pi/3}, e^{-i2\pi/3}\}$ which we can write as $\lhd = e^{i2\pi t/3}$ for $t \in \{-1, 0, 1\}$. Substituting into equation (14), the solution becomes

$$w = \frac{-1}{3c}\left(\frac{\gamma}{\sqrt{\gamma}e^{i(\theta+2\pi t)/3}} + \sqrt{\gamma}e^{i(\theta+2\pi t)/3} + b\right) \tag{19}$$

$$= -\frac{-1}{3c}\left(\sqrt{\gamma}e^{-i(\theta+2\pi t)/3} + \sqrt{\gamma}e^{i(\theta+2\pi t)/3} + b\right) \tag{20}$$

$$= -\frac{-1}{3c}\left(\sqrt{\gamma}\left(e^{-i(\theta+2\pi t)/3} + e^{i(\theta+2\pi t)/3}\right) + b\right) \tag{21}$$

$$= -\frac{-1}{3c}\left(2\sqrt{\gamma}\cos\left(\frac{\theta + 2\pi t}{3}\right) + b\right). \tag{22}$$

3.3. The Algorithm

The derivation of the general solution to Equation (7) is somewhat complex, but the final solution is easy to implement in code and is efficient to compute. Algorithm 1 provides the pseudo code for a function that computes the smallest positive solution to Equation (7) or ∞ if there is no real-valued positive root. The general solver in Algorithm 1 is easily reused to compute the maximum radius for the actual distortion models of interest. For example, Algorithm 2 uses Algorithm 1 to compute the maximum valid radius for the Brown–Conrady model. A similar approach is possible for other polynomial variants up to third degree.

These algorithms allow efficient computation of the maximum valid radius of distortion given the polynomial coefficients. Knowledge of this maximum radius enables two important validation tests in practical applications. First, when estimating a distortion model one can quickly check that the model is bijective for the entire image by simply comparing the maximum radius to the distance to the furthest image corner. This test makes it easy to determine if undistorting the image will produce mirroring artifacts in the corners. Second, for image overlay applications, like augmented reality, the maximum radius can serve as a threshold in the normalized projection space to ensure that scene geometry outside the valid FOV does not erroneously project into the image.

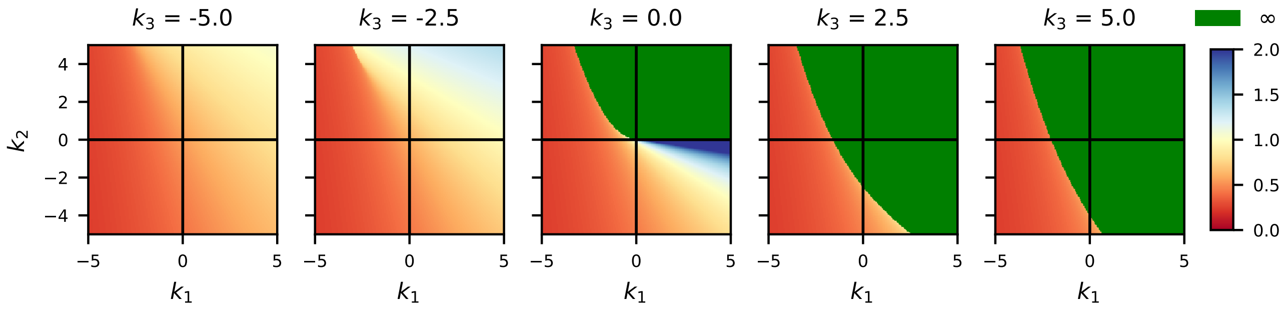


Figure 3: Brown–Conrady maximum valid radius as a function of k_1 and k_2 shown for different values of k_3.

Algorithm 1: Calculate minimum positive root

```
Input: coefficients $\{a, b, c\}$
Output: smallest positive root of $1 + ar + br^2 + cr^3$
        or $\infty$ if no positive root
Function MinPosRoot ($a$, $b$, $c$):
    $r \leftarrow \infty$
    if $c \neq 0$ then
        $\beta \leftarrow 2b^3 - 9abc + 27c^2$
        $\gamma \leftarrow b^2 - 3ac$
        $\Delta \leftarrow \beta^2 - 4\gamma^3$
        if $\Delta > 0$ then
            $U \leftarrow \sqrt[3]{(\beta + \sqrt{\Delta})/2}$
            $r \leftarrow ((\gamma/U) + U + b)/(-3c)$
        else
            $\theta \leftarrow \mathrm{atan2}(\sqrt{-\Delta}, \beta)$
            for $t \in \{-1, 0, 1\}$ do
                $s \leftarrow \left(2\sqrt{\gamma}\cos\left(\frac{\theta+2\pi t}{3}\right) + b\right)/(-3c)$
                if $s < r$ and $s > 0$ then
                    $r \leftarrow s$
    else if $b \neq 0$ then
        $\Delta \leftarrow a^2 - 4b$
        if $\Delta \geq 0$ then
            $\Delta \leftarrow \sqrt{\Delta} - a$
            if $\Delta > 0$ then
                $r \leftarrow \frac{2}{\Delta}$
    else if $a < 0$ then
        $r \leftarrow \frac{1}{-a}$
    return $r$
```

4. Experiments

In this section we summarize how the maximum radius of distortion varies as a function of the polynomial coefficients. We provide an example of how the behaviour of this function can vary when polynomials are fitted to real data. Then we evaluate the impact of the maximum radius of distortion on real camera calibration models used in the wild.

Algorithm 2: Brown–Conrady maximum radius

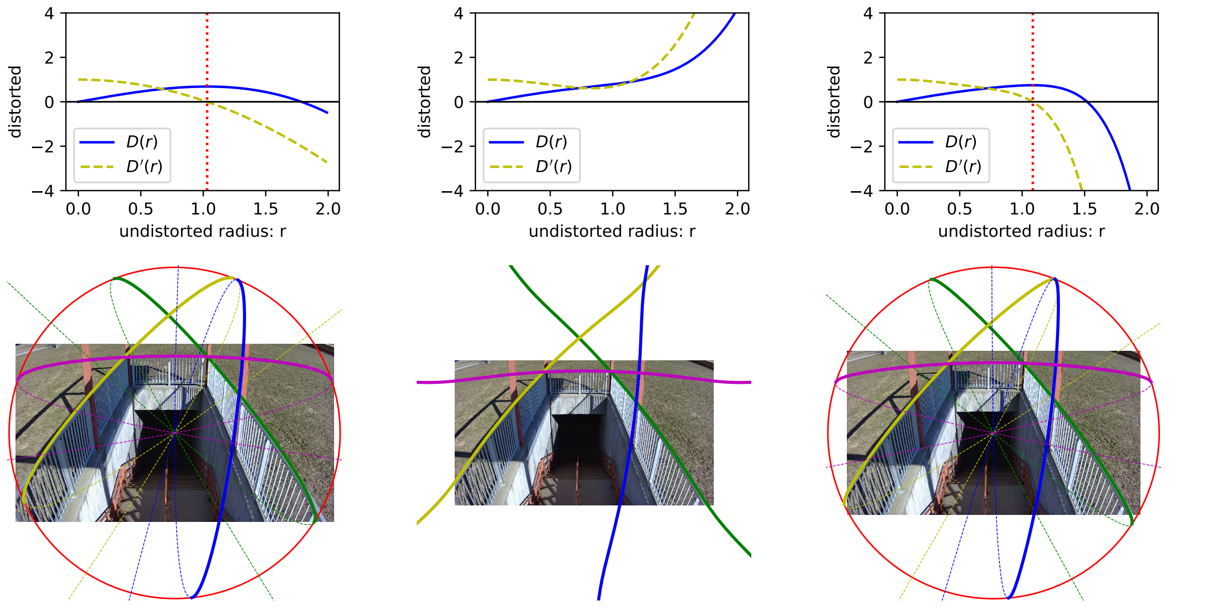
```
Input: coefficients $\{k_1, k_2, k_3\}$
Output: maximum valid radius under distortion
        Equation (4)
return $\sqrt{\texttt{MinPosRoot}(3k_1, 5k_2, 7k_3)}$
```

4.1. Properties of the Maximum Radius

Figure 3 plots the maximum radius of the Brown–Conrady model as a function of the polynomial coefficients. The basic third order model produces a similar plot. Green regions of the parameter space have $r_{\max} = \infty$ and no folding occurs. Blue regions have large but finite radii. Red/orange regions have radii close to zero with danger of folding within the image. As long as all coefficients are positive then we are guaranteed that $r_{\max} = \infty$. Likewise, since the k_3 term dominates, we are guaranteed that $r_{\max} \neq \infty$ if $k_3 < 0$. Notice the sharp transitions between $r_{\max} = \infty$ and $r_{\max} \approx 0$, for positive k_3. This means that small perturbations to the estimates of polynomial coefficients can abruptly transition to from bijective to folded solutions as demonstrated next.

4.2. Effects of Different Order Polynomials

Using the `calibrateCamera` function in OpenCV [21] we fit first, second, and third degree polynomial models to the same calibration data. A Reolink RLC-410S security camera captured the calibration data, a sequence of 35 different images of a circle grid calibration target in different poses and positions within the image. OpenCV uses the Brown–Conrady model. We then deployed this camera in a surveillance setting to capture a stairwell, which conveniently contains many long straight line segments. Figure 4 shows how the best fit model distorts lines for increasing degree polynomials



(a) First degree – max radius: 1.0309
$k_1 = -0.3136$

(b) Second degree – Max radius: ∞
$k_1 = -0.3767, k_2 = 0.1627$

(c) Third degree – Max radius: 1.0855
$k_1 = -0.3902, k_2 = 0.2376, k_3 = -0.1109$

Figure 4: First through third degree Brown–Conrady distortion functions fit with OpenCV to the same data. Top: plots of $D(r)$ and $D'(r)$ for each case with the maximum valid radius marked in red. Bottom: multiple straight world lines distorted under each model. These fold at the maximum valid radius (red circle) in the first and third but not second degree case. All radius units are reported in the undistorted space.

and where the maximum radius of distortion lies. Notice that the solid colored curves (yellow, blue, green, magenta) align with straight objects in each case, but they curve back into the image (dashed lines) in the first and third degree models at the maximum radius (red circle). The same curves extend to infinity in the second degree model even though coefficients have similar values in each case. The first and third degree cases have the maximum valid radius inside the image meaning that the corners are not modeled correctly.

4.3. Maximum Radii of Lenses in the Wild

To study how widespread these issues are in lens models used in production, we used Lensfun [9]. Lensfun is an open source library and database of over 5000 photographic lenses and their properties. Lensfun was developed for photographers to correct image distortions and is used in production software such as Darktable, Rawstudio, ImageMagick, and more. The Lensfun database has also been used, as in this paper, to study lens models in the academic literature [29]. Specifically, we accessed radial distortion models across various consumer lenses to determine how many lenses "in the wild" exhibit this folding problem and how many have folding that occurs within the image bounds.

The Lensfun database contains three different radial lens distortion functions named *ptlens*, *poly3*, and *poly5* defined as

$$F_{\text{ptlens}}(r) = (1 - a - b - c) + cr + br^2 + ar^3, \quad (23)$$

$$F_{\text{poly3}}(r) = (1 - k_1) + k_1 r^2, \quad (24)$$

$$F_{\text{poly5}}(r) = 1 + k_1 r^2 + k_2 r^4. \quad (25)$$

The derivative of the distorted radius for each of these models can be converted to the form of either Equation (3) or (4) by change of variables. For *ptlens* and *poly3*, one must also divide by the constant terms, $1 - a - b - c$ and $1 - k_1$ respectively, to match the scale of Equation (3), where $F(0) = 1$. Scaling does not change the polynomial roots. Thus we seek the roots of

$$D'_{\text{ptlens}}(r) \propto 1 + \left(\frac{2c}{d}\right) r + \left(\frac{3b}{d}\right) r^2 + \left(\frac{4a}{d}\right) r^3, \quad (26)$$

$$D'_{\text{poly3}}(r) \propto 1 + \left(\frac{3k_1}{1 - k_1}\right) r^2, \quad (27)$$

$$D'_{\text{poly5}}(r) = 1 + 3k_1 r^2 + 5k_2 r^4, \quad (28)$$

where $d = 1 - a - b - c$. In these forms one can easily call `MinPosRoot` (Algorithm 1) to solve for the maximum valid radius for each case.

4.4. Quantitative Results

Our theoretical analysis demonstrates that given certain coefficients of the distortion models, lens folding can occur. Our quantitative experiments seek to explore the prevalence of these phenomena in practice. We apply our algorithm to each model in the Lensfun database and compute $r_{\max}$ for each case.

There are several questions we wish to address with this data. The first is how frequently $r_{\max} \neq \infty$, meaning that $D(r)$ is not a bijection for these given parameters and it maps multiple radii in the undistorted space into the same point in the distorted image space. Table 1 shows the number of models with finite $r_{\max}$, broken down by the type of lens model. The primary insight here is that 30.63% of models have $r_{\max} \neq \infty$, leading to non-bijective distortion functions. The low degree *poly3* model is especially prone to this issue.

Model	Finite	Infinite	%Finite	Total
ptlens	1141	3055	27.19%	4196
poly3	411	461	47.13%	872
poly5	2	3	40.00%	5
Total	1554	3519	30.63%	5073

Table 1: Number of lens models with a finite maximum valid radius.

Second, for the cases with a finite $r_{\max}$, we further explore whether the distorted radius function, $D(r)$, tends toward negative or positive infinity. If $D(r)$ tends toward negative infinity, it will eventually pass back through the image, and there will be parts of the world that are outside the camera frustum that incorrectly project back into the image as in Figure 2. Table 2 shows that 98% of finite cases (30% of all Lensfun models) have this issue with distorted radii that tend toward negative infinity. In order to tend toward positive infinity, the remaining 2% must be third degree polynomials (*i.e. ptlens*) with an inflection point greater than $r_{\max}$.

Model	Positive	Negative	%Negative	Total
ptlens	27	1114	97.63%	1141
poly3	0	411	100.00%	411
poly5	0	2	100.00%	2
Total	27	1527	98.26%	1554

Table 2: Fraction of models with a finite max radius that have reprojection functions tending to negative infinity.

Third, For the cases with a finite $r_{\max}$, we compared the finite maximum radii to the image sizes to validate that the Lensfun models are at least valid for the entire image domain. Otherwise, these models would fail at their intended application of undistorting images. The `MinPosRoot` algorithm returns the maximum valid radius in the normalized undistorted image space. We map $r_{\max}$ to the normalized distorted image space, $D(r_{\max})$, and compare to the image corner radius after normalizing for focal length and image scaling. In these experiments, we only use the lens models marked as *rectilinear* in the Lensfun database, ignoring fisheye and other projections. Rectilinear covers 1532 out of the 1554 models that have a finite maximum radius.

To obtain the normalized radius of the image corners, we need the focal length, f, and the scale of the crop relative to the reference sensor, s_c. Both of these parameters are taken directly from the Lensfun database. Lensfun uses a reference sensor size of 36mm $\times$ 24mm. To obtain the distance to the corner in the image space, the distance from the center of the sensor to the corner is computed and divided by the focal length and crop scale as follows

$$r_{\text{corner}} = \frac{\sqrt{18^2 + 12^2}}{s_c f}. \tag{29}$$

We now define a normalized radius representing how large the max radius is relative to the corner of the image

$$r_{\text{normalized}} = \frac{D(r_{\max})}{r_{\text{corner}}} \tag{30}$$

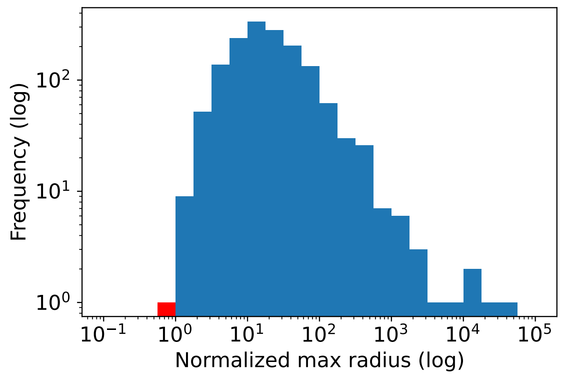
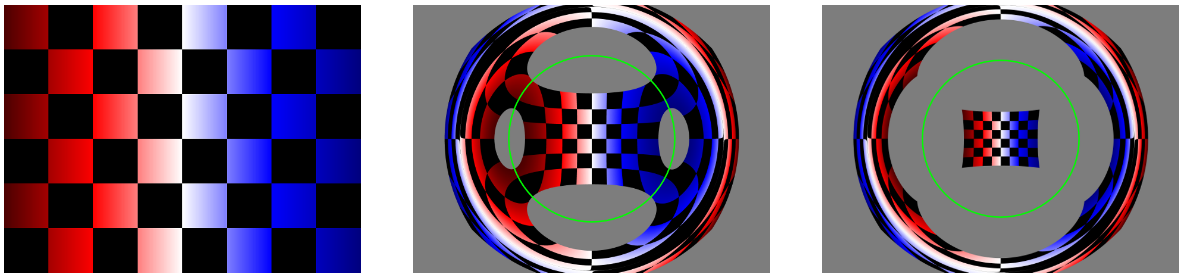


Figure 5: Histogram of normalized finite maximum radii, $r_{\text{normalized}}$, on a log-log plot. Only one model, corresponding to Figure 6b, has $r_{\text{normalized}} < 1$, meaning folding occurs *within the bounds* of the input distorted image. Note that most other models still have problematic distortion in the case of forward projection, as shown in Figure 2.

If the value of $r_{\text{normalized}}$ exceeds one then the maximum valid radius lies outside of the image bounds and the model is valid for the entire image. If less than one, the model is not a bijection over the image domain and warping to remove distortion will have artifacts. Figure 5 plots a histogram showing the distribution of $r_{\text{normalized}}$ values across the 30% of the Lensfun database with finite values. Only


(a) Input checkerboard used to visualize undistortion functions.

(b) Undistortion for a Tokina AF AT-X Pro DX, at a focal length of 11mm. $r_{\text{normalized}} = 0.858$ is shown in green.

(c) Undistortion for a Sigma 14mm f/1.8 DG HSM | A. $r_{\text{normalized}} = 1.478$ is shown in green.

Figure 6: Visualizations of undistorting a synthetic colorized checkerboard using problematic models from the Lensfun database. Since we do not have access to captured images from these lenses, we use a checkerboard pattern (a) simply for visualization. The one case where $r_{\text{normalized}} < 1$ is in (b). An example where $r_{\text{normalized}} > 1$ is in (c).

one camera model has $r_{\text{normalized}} < 1$. In this case it is impossible to undistort that image and capture all the corners without including problematic regions as shown in Figure 6b. It is unsurprising that nearly all of the Lensfun models are free from this issue because they are generally manually vetted by visualizing undistorted images before models are added to the database. It is expected that the occurrence of estimated models with this issue is much higher before manual vetting.

In general, these folding artifacts occur after a gap outside the image, which can still cause erroneous projections when overlaying projected 3D geometry onto the distorted image. An example with the maximum valid radius just outside the image can be seen in 6c where a spurious ring occurs around the correctly undistorted internal image. This problem is especially severe because, as shown in Table 2, the vast majority of distortion functions with a finite maximum radius tend to negative infinity. All functions with the property and $r_{\text{normalized}} > 1$ will project all points in the distorted image into three points in the undistorted image as seen in Figure 6c, once in the correct location, once further out, and once even further from the image center on the opposite side.

5. Conclusion

This paper has documented some of the often overlooked challenges of working with polynomial radial distortion that are prevalent in computer vision applications. Specifically, we showed that these models can, in some cases, fold back on themselves erroneously mapping points from outside the field of view into the image bounds. Distortion models do not always fold, but folding is far from a rare occurrence with 30% of models in the Lensfun database observed to have this issue. We derived a closed form solution for determining the maximum valid radius. This maximum radius is easily computed and can serve as a threshold in augmented reality applications to filter invalid points during rendering. The algorithm has been contributed to KWIVER [12, 18], a large open source C++ computer vision code base, to serve exactly this purpose. Furthermore, the maximum distortion radius can be compared to image maximum radius as a validation test to ensure that estimate distortion models are valid for the entire image.

It is possible to use other heuristic thresholds to avoid invalid projections. For example, one can map the corner radius of the image into the undistorted space and use this as the threshold. However, this will also filter many other points that fall outside the image and will not properly represent the case where folding happens within the image bounds. In some applications the projection of content that lies outside the image can provide additional context. Using the maximum radius of distortion allows for the most context possible across the domain where the lens model is valid.

There are other applications of the maximum distortion radius to explore in future work. Most notably we will explore the use of the maximum distortion radius as a constraint in the optimization of polynomial coefficients. It should be possible to estimate polynomial coefficients under the constraint that the maximum distortion radius is infinite, or that it is greater than a selected target radius.

Acknowledgement

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via contract 2017-16110300001. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA or the U.S. Government.

References

[1] Luis Alvarez, Luis Gómez, and J Rafael Sendra. An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision*, 35(1):36–50, 2009.

[2] João Pedro Barreto and Kostas Daniilidis. Fundamental matrix for cameras with radial distortion. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 625–632. IEEE, 2005.

[3] Christian Bräuer-Burchardt. A simple new method for precise lens distortion correction of low cost camera systems. In *Joint Pattern Recognition Symposium*, pages 570–577. Springer, 2004.

[4] D Brown. Decentering distortion of lenses. *Photogrammetric Engineering.*, 32(3):444–462, 1966.

[5] Faisal Bukhari and Matthew N Dailey. Automatic radial distortion estimation from a single image. *Journal of mathematical imaging and vision*, 45(1):31–45, 2013.

[6] David Claus and Andrew W Fitzgibbon. A rational function lens distortion model for general cameras. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 213–219. IEEE, 2005.

[7] A. E. Conrady. Decentred Lens-Systems. *Monthly Notices of the Royal Astronomical Society*, 79(5):384–390, 03 1919.

[8] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *Journal of Mathematical Imaging and Vision*, 40:123–148, 2000.

[9] Lensfun developers. Lensfun. `https://lensfun.github.io/`, 2021.

[10] Frederic Devernay and Olivier Faugeras. Straight lines have to be straight. *Machine vision and applications*, 13(1):14–24, 2001.

[11] William Dunham. Cardano and the solution of the cubic. In *Journey through genius: The great theorems of mathematics*, chapter 6, pages 133–154. Wiley, 1990.

[12] Keith Fieldhouse, Matthew Leotta, Arslan Basharat, Russell Blue, David Stoup, Charles Atkins, Linus Sherrill, Benjamin Boeckel, Paul Tunison, Jacob Becker, Matthew Dawkins, Matthew Woehlke, Roderic Collins, Matt Turek, and Anthony Hoogs. Kwiver: An open source cross-platform video exploitation framework. pages 1–4, 10 2014.

[13] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR*, volume 1, pages I–I. IEEE, 2001.

[14] David A Forsyth and Jean Ponce. *Computer vision: a modern approach.* Pearson, 2012.

[15] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1309–1321, 2007.

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[17] Richard I. Hartley and Tushar Saxena. The cubic rational polynomial camera model. In *In Image Understanding Workshop*, pages 649–653, 1997.

[18] Kitware Inc. KWIVER v1.6.0. `https://github.com/Kitware/kwiver/blob/v1.6.0/vital/types/camera_intrinsics.cxx#L301`, 2021.

[19] David E Muller. A method for solving algebraic equations using an automatic computer. *Mathematical tables and other aids to computation*, 10(56):208–215, 1956.

[20] OpenCV Forum. Undistortion at far edges of image. `https://answers.opencv.org/question/28438/undistortion-at-far-edges-of-image/`, 2014.

[21] OpenCV team. OpenCV: Open source computer vision library. `https://opencv.org/`, 2021.

[22] Srikumar Ramalingam, Peter Sturm, and Suresh K Lodha. Generic self-calibration of central cameras. *Computer Vision and Image Understanding*, 114(2):210–219, 2010.

[23] Robotics StackExchange Forum. Stereo camera calibration with different camera types. `https://robotics.stackexchange.com/questions/11106/stereo-camera-calibration-with-different-camera-types`, 2016.

[24] Stack Overflow. Python 2.7/opencv 3.3: Error in cv2.initundistortrectifymap. not showing undistort rectified images. `https://stackoverflow.com/questions/46100927/python-2-7-opencv-3-3-error-in-cv2-initundistortrectifymap-not-showing-undist`, 2017.

[25] Gideon P Stein. Lens distortion calibration using point correspondences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 602–608. IEEE, 1997.

[26] Rickard Strand and Eric Hayman. Correcting radial distortion by circle fitting. In *Compuational Vision and Active Perception Laboratory*, 01 2005.

[27] Rahul Swaminathan and Shree K Nayar. Nonmetric calibration of wide-angle lenses and polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1172–1178, 2000.

[28] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[29] Zhongwei Tang, Rafael Grompone von Gioi, Pascal Monasse, and Jean-Michel Morel. A precision analysis of camera distortion models. *IEEE Transactions on Image Processing*, 26(6):2694–2704, 2017.

[30] Thorsten Thormählen, Hellward Broszio, and Ingolf Wassermann. Robust line-based calibration of lens distortion from a single view. *Mirage 2003*, pages 105–112, 2003.

[31] Roger Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[32] Aiqi Wang, Tianshuang Qiu, and Longtan Shao. A simple method of radial distortion correction with centre of distortion estimation. *Journal of Mathematical Imaging and Vision*, 35(3):165–172, 2009.

[33] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.