A Non-iterative Method for Correcting Lens Distortion from Nine Point Correspondences

Hongdong Li and Richard Hartley

Research School of Information Sciences and Engineering,
Institute of Advanced Studies, The Australian National University
ASST, Canberra Labs, National ICT Australia Ltd.

Abstract

This paper presents a new method for calibrating and correcting large radial distortion. It makes use of a number of image point correspondences from two views only. No knowledge of the scene structures, nor camera intrinsic parameters, is required. By using two singularity conditions, the method successfully decouples the estimation of the radial distortion from the estimation of fundamental matrix. The solution technique is basically non-iterative, it thereby does not need any initial guess, with no risk of local minima. It also proposes a kernel-voting scheme (instead of the conventional RANSAC scheme). The result is shown to be reliable and robust to noise. In addition, the method is easy to implement.

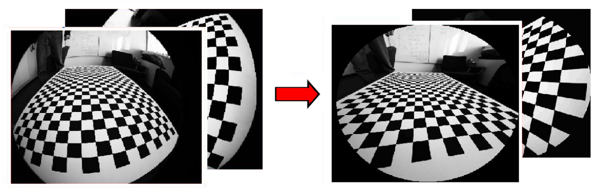

Figure 1: Demo of our method. Input: two views with significant radial distortions, find image point correspondences. (Here the **unknown** 3D scene points can be coplanar or not); Output: centre-of-distortion, distortion parameters, and the corrected images.

1 Introduction

This paper presents an easy method for calibrating and correcting large radial lens distortions. Radial distortion is a significant problem in the analysis of digital images. It is very common for wide-angle camera, fisheye camera, catadioptric camera and those cheap cameras with short focal-lengthes. Although this problem was widely studied by photogrammetrists, striving for extreme accuracy, it has been largely overlooked in the extensive literature of structure-from-motion (SfM) during the past decade or so. Using a radially mis-aligned image in a SfM algorithm may cause significant skewness [4].

The classic methods for camera geometric calibration make use of a carefully manufactured calibration grid. And almost exclusively, lens distortion parameters, as well as other intrinsic/extrinsinc camera parameters are estimated in a single optimization framework at the same time. Such nonlinear iteration can be troublesome, due to lack of convergence, choosing an initial estimation, local minima, and determining a stop criterion. In addition, employing such classic method is very laborious. Zhang in his *flexible calibration* work based on a planar calibration grid also incorporated the estimation of radial distortion [3].

Many papers have been devoted to the so-called *plumb line* idea which utilizes the fact that straight line (as a whole) is an invariant entity under the ideal pinhole projection [21]. In fact, any single-view projective invariant can be plugged into this scheme [14]. The difficulties with this kind of method is that a knowledge of the scene must be known in advance, and the particular image object corresponding to this knowledge must be identified as well. For example, in the *plumb line* algorithm the user must tell computer which curve is actually the image of a straight line. Some semi-automatic approaches have been introduced therein such as the *radial distortion snake* [22]. In this paper we attempt to only use image point correspondences to calibrate the radial distortion. The 3D points can be coplanar or not. A sample scenario of applying our method is shown in figure-1. This method can be roughly thought as an auto-calibration technique. Zhang has studied this idea, and proposed a generalized two-view epipolar relationship with lens radial distortion [2]. He suggested using *bundle adjustment* to solve for the distortion. Stein proposed a method to calibrate lens distortion from point correspondences of two view case and three view cases [13]. Both methods need to solve an enlarged camera calibration problem where the unknown set also includes the distortion parameters. Usually an iterative optimization algorithm (bundle adjustment) is employed. Fitzgibbon introduced an interesting method for simultaneously estimating radial distortion and fundamental matrix [4]. He formulated the problem in a Quadratic Eigenvalue Problem (QEP) form and applied available numerical technique to solve it. His framework works only for one-parameter radial distortion model. Micusik and Pajdla have extended this idea to a two-parameter model for fisheye-lens, but after approximation they essentially solve a similar one-parameter problem [7].

We already mentioned some drawbacks with *nonlinear iteration*. Here, we point out another one issue, which is more serious: experiment results have shown that there is certain kind of *coupling* or *correlation* among camera parameters, which could make the estimation result rather unreliable (see [24]). We show in this paper that our method has successfully **decoupled** this coupling. It therefore yields more stable and more reliable estimation. Our method bases itself on some non-linear equations of singularity, but the solution techniques it applied are non-iterative.

To utilize multiple noisy measurements, we propose a scheme of *kernel-voting* which proves to be robust to noise, and is more applicable than the conventional RANSAC in the problem context. We have obtained good results. Our method is generic enough in that in principle it allows for any algebraic parametric distortion model. We do not make any assumption about the form of the distortion model except for being **algebraic**.

Our work benefits from the following very recent results. Thirthala and Pollefeys proposed a method called *radial trifocal tensor* [5]; Claus and Fitzgibbon presented a Rational Function Model applying to non-linear lifted image correspondences [6]. They have successfully rectified (straightened) the epipolar lines, yet some ambiguities in extracting distortion parameters remain; Hartley and Kang provided a model-free model for radial distortion correction. They also gave a novel algorithm to estimate centre-of-distortion [9].

2 Radial distortion models

This section explains several commonly adopted radial distortion models that are most related to our work described here. We already mention that our method does not rely on particular model. However, a minimal requirement is that the model must be (elementary) **algebraic** (rather than transcendental). Therefore, we will not consider here the FOV model [21], nor the model used by Micusik and Pajlda[7], because they contain trigonometry functions.

Polynomial Model (PM) is the most popular model to describe radial distortion:

$$\mathbf{x}_u - \mathbf{e} = (\mathbf{x}_d - \mathbf{e})L(r_d, \mathbf{k}), \qquad (1)$$

where

$$L(r_d, \mathbf{k}) = 1 + k_1 r_d^2 + k_2 r_d^4 + \cdots + k_p r_d^{2p}, \qquad (2)$$

and $2p$ is model order, $\mathbf{e}$ the centre-of-distortion (COD) and r_d the pixel radius to $\mathbf{e}$.

PM model works best for lens with small distortions. For wide-angle lens or fish-eye lens that have large distortion, it often require too many terms than practical.

Fitzgibbon suggested the use of Division Model (DM):

$$\mathbf{x}_u - \mathbf{e} = (\mathbf{x}_d - \mathbf{e})/L(r_d, \mathbf{k}), \qquad (3)$$

where $L(r_d, \mathbf{k})$ is the same in (2). The most remarkable advantage of DM over PM is that it is able to express high distortion at much lower order. In particular, for many cameras one parameter suffices [4][6].

Of course, combining (3) and (1) we can get a more generic Rational Model (RM):

$$\mathbf{x}_u - \mathbf{e} = (\mathbf{x}_d - \mathbf{e})\frac{L_1(r_d, \mathbf{k}_1)}{L_2(r_d, \mathbf{k}_2)}, \qquad (4)$$

This RM should not be confused with [6]'s Rational Function Model, nor with [8]'s rational cubic model, as the former applies to lifted coordinates, and the latter encapsulates all imaging process including the projective transformation. One favorable property of the above models (to our method) is that they are all elementary algebraic models.

There are other radial distortion models designed for other non-traditional camera using curve-mirror. For example, [17],[23],[15],[20],[14]. Since they are mostly algebraic, our method can be easily adapted to these novel models without effort, yet we will not discuss them in this paper.

3 Find COD using radial fund-matrix

The main correction algorithm proposed in this paper relies on an accurate estimation of the centre-of-distortion (COD).

While it is common in the literature to assume the COD is known, usually assumed at principal point, we argue this is not a safe assumption in general. In this paper, we make no assumption about the position COD. Instead, we actually *estimate* it. Traditionally, the actual estimation of the COD is obtained at the same time of performing a full-scale camera calibration, which is often a tedious procedure. Micusik and Pajdla suggested using the center of the circular field-of-view as the COD, but it only works for the situation when the whole (circular) field-of-view is seen in full in one image [7].

We adopt a new method proposed in [9]. This method is simple, yet produces good result. Here we briefly sketch this method. For more details the reader is referred to [9]. We assume the radially distorted image have all square pixels, i.e., the aspect ratio is unity. Now, let the camera observe a planar scene with known coordinates, for example a planar checkerboard calibration pattern (e.g., fig-1). Given $\mathbf{x}_c$ as known point on the planar calibration pattern, and $\mathbf{x}_d$ the corresponding image point in the distorted image. These two points are related by a so-called radial fundamental (epipolar) relationship, which can be written as:

$$\mathbf{x}_d^T \mathrm{F}_\mathbf{r} \mathbf{x}_c = 0,$$

where the matrix $\mathrm{F}_\mathbf{r}$ is called as **radial fundamental matrix** by [9], and its formal mathematical derivation is also given there. The matrix may be computed in the usual way (for example, the eight point algorithm) from several point correspondences, and the COD extracted as the **left epipole**:

$$\mathbf{e}^T \mathrm{F}_\mathrm{r} = 0.$$

This method can be extended to image with non-square pixels and to unknown planar scene. For notation's sake, in the remaining parts of the paper, where there causes no confusion, we simply assume that the COD $\mathbf{e}$ has already been estimated and subtracted from the point coordinates $\mathbf{x}_u$ and $\mathbf{x}_d$.

4 Basic idea: nine-point algorithm

Consider two views of a static scene. Let ${\mathbf{x}_u}'$ and $\mathbf{x}_u$ denote a pair of correspondences, of the two undistorted images, respectively. The epipolar (coplanar) relationship is written as:

$${\mathbf{x}_u}'^T \mathrm{F} \mathbf{x}_u = 0, \qquad (5)$$

where matrix F is fundamental matrix (or essential matrix if the camera is intrinsically calibrated).

Assume image pixels are all square (i.e.,zero-skew and unity aspect-ratio). Now plugging any (algebraic) radial distortion model into it, we thus get a generalized epipolar equation, which explicitly depends on the radial distortion. For example, using (3) in (5) we get

$$[{\mathbf{x}_d}'/L({r_d}', \mathbf{k})]^T \mathrm{F} [\mathbf{x}_d / L(r_d, \mathbf{k})] = 0. \qquad (6)$$

Note that the image coordinates being used are homogeneous, they are thereby admit arbitrary change in scale without changing the equity of the equation. We thus multiply the $L(r_d, \mathbf{k})$ on both sides of the left-term of the equation, and rearrange it in a bilinear form of the homogeneous coordinates components (x, y, z), using Kronecker product symbol $\otimes$, so get

$$(({x_d}', {y_d}', L({r_d}', \mathbf{k})) \otimes (x_d, y_d, L(r_d, \mathbf{k})))\mathtt{vec}(\mathrm{F}^T) = 0.$$

Now we do so for nine points, whose coordinates denoted by matrices X' and X, and then stack the nine bilinear equations together, thus get a homogeneous equation system:

$$\mathrm{M}(\mathrm{X}', \mathrm{X}, \mathbf{k})\mathbf{f} = 0, \qquad (7)$$

where the square matrix M is called *measurement matrix*, which depends explicitly on input distorted coordinates and the distortion parameter $\mathbf{k}$, $\mathbf{f}$ the right null vector. For simplicity, later we will drop the X and X' in M.

We then make two important observations: firstly, we find that the $\mathbf{f}$ is nothing else but the $\mathtt{vec}(\mathrm{F}^T)$. This is because the row-wise re-scaling of M does not affect its null-space at all; secondly, this row-wise re-scaling does not change its rank either.

This homogeneous equations will have non-trivial solution *iff* matrix $\mathrm{M}(\mathbf{k})$ is singular. Moreover, since its solution is a valid fundamental matrix, so it itself (after rearranging) must be singular too. Writing down these two singularity conditions, we then get a pair of nonlinear equations.

$$\det(\mathrm{M}(\mathbf{k})) = 0 \qquad (8)$$

$$\det(\mathtt{Mtx}[\mathtt{Ker}[\mathrm{M}(\mathbf{k})]]) = 0, \qquad (9)$$

where the `Ker[]` is the null-space operator, and `Mtx[]` is the *matrix operator* which rearranges a vector into a matrix.

These two singularity conditions are well-known in vision geometry research, but to the best of our knowledge they have never been used for such problem. They play a central role in this paper, and therefore are called as *basic equations*. Note that the distortion parameter only depends on the singularity conditions, and has little to do with the entry values of the F. We therefore successfully decouple the estimation of distortion from the fundamental matrix. Consequently, our method works equally well for calibrated (in the usual sense) and un-calibrated camera.

Now that having a group of nine correspondences, two nonlinear basic equations are established. If the distortion model makes use of two parameters, then nine points are sufficient to estimate them. When more parameters are used, in principle we may simply collect more groups of measurements and then solve the resultant equations.

4.1 An example: One-parameter problem

In this section we demonstrate our method (and nine-point algorithm) to the one-parameter DM model, i.e, $\mathbf{x}_u = \mathbf{x}_d/(1 + kr_d^2)$. This is by no means a *toy* problem, because all the basic operations applied here can be adapted to the multi-parameter case in a similar fashion. The DM model not only has pedagogical meaning, but also is of practical significance. It is shown that in practice the DM model has much richer expressing power in describing large distortion than the PM model does ([6]).

Since we use algebraic function to describe radial distortion, the basic equations are thereby algebraic in the unknown k. This facilitates the application of various algebraic nonlinear equation solution techniques, for example, companion matrix technique, or Sturm bisection technique, etc [12]. Here we wish to avoid the use of Newton iteration, nor Homotopy, as the former requires good initial estimation and the convergence is not always guaranteed, and the later is often subject to numerical unstable.
Our nine-point algorithm goes as follows.

1. Input two images; find image point correspondences.
   -This can be done by a Harris corner detector followed by a correlation matching algorithm. The point coordinates are required to reach sub-pixel precision.

2. Normalize image coordinates by scaling them using an isotropic scale factor, so that the maximal radius (with respect to COD) is 1.00.
   -This normalization is very crucial for the success of the algorithm, as high-degree nonlinear equations are involved here. Without good conditioning, the final result would be far from correct. A further remark about normalization is that: there is an inherent ambiguity in estimating distortion parameter and magnification parameter (i.e., the focal length). In particular, the change of the scale of the distortion parameter can be absorbed in the change of focal length. In this paper, we will not always enforce the correct scaling condition, but will correct the overall scale at the end of the whole process.

3. Collect a group of nine points, write down the pair of basic equations.
   -Because there is only one unknown k that needs to estimate, in our implementation we only use the first basic equation, also because it has lower total degree than the second. Note that in matrix M the k appears only in five columns, thus a six-degree univariate polynomial in k of the first equation is obtained.

4. Solve this six-degree polynomial equation by *companion matrix* method.
   -We choose this method mainly for its linearity and simplicity. The notion of *companion matrix* is simple: the roots of a monic polynomial equation of $p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ are simply the eigenvalues of its companion matrix:

$$\mathcal{C}_{p(x)} = \begin{pmatrix} 0 & 0 & \dots & \dots & \dots & -a_0 \\ 1 & 0 & \dots & \dots & \dots & -a_1 \\ 0 & 1 & \dots & \dots & \dots & -a_2 \\ 0 & 0 & \ddots & & & \vdots \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & \dots & 1 & -a_{n-1} \end{pmatrix}$$

   In matlab, the command for computing companion matrix is `compan(p)`, and there are plenty of efficient algorithms for eigen-decomposition. The reader may argue that there may involves a hidden non-linear iteration that is used by an eigen-decomposition procedure, there is, however little risk in choosing initial guess, or local minima.

5. In general, we will always get six complex roots, but only need to keep the real ones. However, there could be still more than one, or even no real soot. For either case, we need to use multiple measurements, solve them and then single out the *best root*.
   -By the *best root* we mean the one that is consistent with most measurements. A possible way to choose it is to test this root against all other measurements by, for example, a RANSAC technique. However, we argue that RANSAC is not the best suited technique for the underlying problem. In next section we will propose an alternative and more efficient technique.

6. Output the best root and end.

5 Using more measurements: voting on real roots

Using more measurements in general increases the stability of computation. If we use N ($N \gg 1$) groups of correspondences, we then end up with a system of N simultaneously nonlinear equations. Physically, even though the true k should satisfy all these equations, due to noise the obtained simultaneous equations can hardly find any consistent solution. In other words, the system of equations has no solution.

The last step of the above nine-point algorithm is to use RANSAC to find a most consistent real root. But RANSAC is inefficient in such problem context. The reasons are:

- firstly, unlike in the problem of estimating a line or a fundamental matrix where the inlier-outlier-test can be performed fairly efficiently, for radial distortion estimation there is not simple way to do so. If one insists in using RANSAC, he has to first tentatively undistort the image using the current distortion parameter estimation, then compute the fund-matrix and reprojection error, and count inliers/outliers. This approach is a bit of over-kill, and does not thoroughly decouple the distortion form the fundamental matrix.

- Secondly, because noise also affects the nine-point group, it thus consequently **distorts** the basic equations as well, as what we handle here are all polynomial equations of high-degree which are in general not so stable. In other words, the equation we just solved may be not the equation that we intended to solve. In such case there is little hope to obtain a genuine root from RANSACing.

To overcome this problem, we propose a kernel-voting scheme. By experimentations, we found: although noise affects the basic equations significantly, the solved roots actually all surround the genuine root. The distribution of all roots from multiple measurements shows a peak shape. So long as we collect enough measurements, an asymptotically-correct root will be eventually found. Our simulations show that this number needs not to be very large. Usually $30 \sim 70$ suffices. Another benefit of the voting scheme is that it is very robust to outliers. A similar theoretical analysis of the success-rate (probability) as used in the RANSAC scheme could be carried out.

In the voting scheme, we apply a kernel density estimator (KDE) to find the position corresponding to the (globally) maximal (peak) probability. This peak detection task could also be done by a simple histogram technique. But with histogram there is the difficulty of determining bin numbers. Another possible way is to use median position, but it lacks theoretic justification.

The goal of density estimation is to approximate the probability density function of random variables. Assume we have independent observations from the random variable. The Kernel Density Estimator (KDE) for the estimation of the density value at point is defined as

$$\hat{f}_h(x) = \sum_{i=1}^{n} \frac{K(x_i - x)}{h} \tag{10}$$

where $K()$ denoting a so-called kernel function, and h the bandwidth. Here we choose a Gaussian kernel with fixed bandwidth. Basing on the estimated distribution density of real roots, we then easily identify the root position corresponding to the largest peak of the density function. Experiments show that the precision is good(see figure-4).

6 Multi-parameter problem

Following the same algebraic fashion, our algorithm can be extended naturally to multi-parameter case. Now we give some thoughts and preliminary experiment for this issue. As an example, we study the two-parameter DM case (after COD removal),i.e, $\mathbf{x}_u = \mathbf{x}_d/(1 + k_1 r_d^2 + k_2 r_d^4)$.

Collecting a group of nine points, we find equation eq. (8) is a two-variable polynomial consisting of the following 28 monomials:

$$\{k_1k_2^2, k_2^2, k_1^2k_2, k_1^2k_2^3, k_1^4, k_1^3k_2, k_1k_2^4, k_1^2, k_1, k_1^3,$$
$$k_2^3, k_2^5, k_1^2k_2^2, k_1^3k_2^2, k_1^4k_2^2, k_2^5, k_1^2k_2^4, k1^3k_2^3, k_1^5k_2,$$
$$k_1k_2, k_1^4k_2, k_1k_2^3, k_2^4, k_1^5, k_2^6, k_1^6, k_2, 1\}.$$

For this particular **bi-variate** problem, one direct way to solve it is by plane-curves-intersecting. Regarding the two basic equations, eq. 8 and eq. 9 as two plane algebraic curves in the k_1k_2 plane (,we are able to do so because the basic equations are all in real coefficients), then the intersection points must be the common *real* roots that we are after.

Alternatively, we propose another approach. Since we do not want to use eq. 9 because it involves higher degree polynomial, while eq. 8 still remains degree-six for each of the variables. Therefore, our strategy is to collect enough groups of data, and obtain enough number of equations, then form a system of equations in the variable k_1 and k_2. In principle, Gröbner basis method can be applied here to generate new equations, but it is no need here because we may have enough linearly independent equations simply by collecting *sufficient* measurements. Reducing this equation system using a modified Gaussian-Jordan Elimination method similar to [11] and [12], we again get an univariate polynomial in k_1. Applying the same method of companion matrix and root-voting described in sec. 4 and 5, we then find the best k_1. Substituting it back, we may find k_2 too.

This procedure can be further extended to cases with more than two parameters. For solving a nonlinear system with multiple unknowns, various type of *resultant* methods can be used here. Detailed explanations for handling such multi-parameter case will be reported separately. It is however worth noting, as the increase of number of parameters, the total degree and the number of terms of the resulting nonlinear equation also increases quickly, and this will have some practical problems. Fortunately, for the radial distortion problem, $1 \sim 4$ parameters usually suffice.

The reader might have thought that our method in essence is very similar to Fitzgibbon's [4]. However, even for the one-parameter case, that method attempts to simultaneously solve both fundamental matrix and distortion, while ours spares the unnecessary computation of the fundamental matrix.

7 Experiments

We give some experimental results in this section, to show the effectiveness and efficiency of the proposed method.

7.1 Tests on synthetic images

We generated a 3D points scene, where the points uniformly randomly distribute within a cube. Then, perspectively projected them on two image planes with different poses and positions, and then applied the radial distortion. The obtained image size is of about 256×256. We used one-parameter DM model with known parameter k to synthesize the distortion. We tested our method for different values of k. After the above procedures, we added Gaussian noise to the image points coordinates. Figure-2 left is the simulated 3D scene, and the right is the corresponding 100 feature correspondences of the two views. the image size is about 256×256.

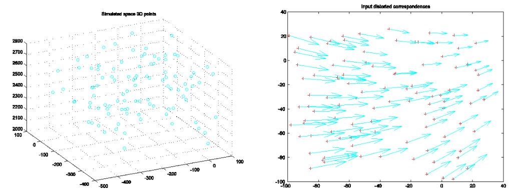

Figure 2: (Left) 3D points scene used for simulation. (Right) distorted image correspondences (100 points). Image size is about 256x256.

Random choosing groups of measurements from the 100 samples, each group has 9 points, we then applied our nine-algorithms to each group. Here we show an example of the six-degree polynomial, just to get a flavor:

$$p(k) = -0.435e^{-18}k^6 - 0.306e^{-15}k^5 + 0.104e^{-13}k^4 - 0.968e^{-13}k^3 + 0.294e^{-12}k^2 - 0.117e^{-11}k + 0.328e^{-11}.$$

It is worth stressing again that: due to the effect of noise, solving any single of the resulting equation in general will not give a correct result. The reader may verify this using the above polynomial. By using a Gaussian kernel, we performed a kernel density estimation on all the solved real roots. In our experiment we had sampled 50 data groups, so we got in total 50 polynomial equations.

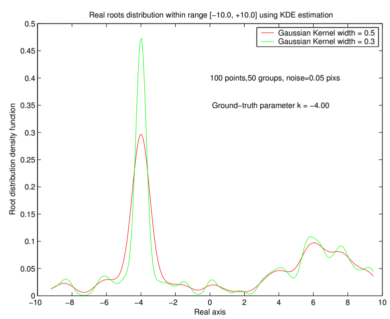

Figure 3: Root voting result: KDE estimation of the density of real roots computed from 50 data groups. Red: Gaussian kernel width=0.5, Green: Gaussian kernel width=0.3, noise level=0.05 .

The resulting density function is depicted in figure-3. It is the average of 200 random tests. The noise level was 0.05 pixels. For this we can easily read out the root value at the peak position of the estimated density function, which is $k = -4.0000127521$, while the ground-truth value is $k = -4.000$.

Zhang [2] ever observed an interesting phenomenon that if distortion is small, his method may not give accurate estimation. We specially test this issue by simulating a very small distortion. Our results (average of 20 tests,under

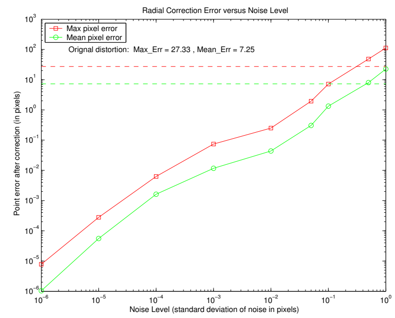

Figure 4: Distortion removal precision versus various noise levels applied to synthetic data. Here we show the maximal pixel deviation and mean deviation away from the ideal position by a pinhole camera.

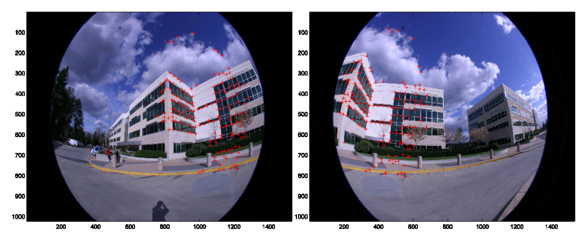

Figure 5: Sample input real images, with extracted feature points superimposed on.

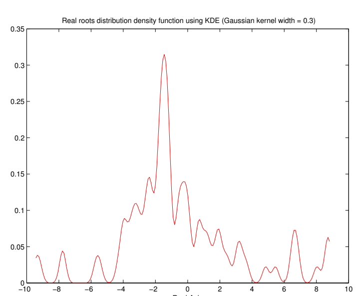

Figure 6: KDE estimated real roots density function. From this we can read out the best root as $k = -1.475$.

0.01 pixels noise) are given in the table below, which are satisfying.

true-k	0.0	-1e-5	-1e-3	-1e-1
est-k	-0.152e-14	-0.997e-5	-0.989e-3	-1.021e-1

We tested the performances of our algorithm under different levels of image noise. Figure-4 gives the correction precision vs. different noise levels. Note that the synthetic image coordinates were in the range [-128, +127], say, image size is by 256 pixels. In the original distorted image, the largest pixel displacement (from the ideal position by a pinhole camera) is about 27.33 pixels, and the mean displacement is 7.25 pixels. When noise level is below 0.5 pixels, the distortion-correction procedure gives positive correction result. In fact, in real image experiments using Harris corner detector with sub-pixel precision one can easily reach < 0.1 pixel precision in a 256×256 image.

We used MATLAB Symbolic-Math Toolbox for all the computations. The total computation (for 50 groups) costs about 15 seconds on a moderate P4@1.8GHz machine. This timing can be easily and substantially reduced, as in our implementation we have paid no attention to code optimization.

7.2 Tests on real images

We tested our algorithm on three different cameras, a Canon-EOS with fisheye lens (image resolution 1536×1024), a Flea-1394 Camera of PtGrey (resolution of 1024×768) and an Hitachi-DZMV580 video camera. The below is a sample image pair by the Canon-EOS, with the detected feature points superimposed on (fig-5). The feature points were extracted by a SIFT detector. Then, we manually found the matches between the two images. After applied our method (i.e., section 4.1 and section 5 after COD removal), we got the following roots distribution density function (fig-6, from which we read the distortion parameter is $k = -1.47523$. We also tried the two-parameter DM model on the same real image pair, the estimation result is $k_1 = -2.2651, k_2 = 1.6282$. Though we have not quantitatively compare the correcting results by these two models, but from the resulting images we can see both produce good distortion removals.

The resulting distortion-corrected images are shown in figure-7. We tested the reliability of our method against noise and outliers (mismatches). In the noise experiments, we rounded the corner coordinates up to **integers** and again added into some extra Gaussian noise of different levels (up to std of 0.2 pixels), then checked the variation of the estimated parameter. We found this only introduced a smoothing-effect in the density curve, but little change to the peak position. In another outlier experiment, we arbitrarily added a small numbers of mismatches into the input data, then ran our algorithm again. We found although the roots became more scattering, the peak position remained stable. Moreover, by increasing the sampling number of groups, we found the peak can be made as sharp as the outlier-free case. We tested our method on a Point-Grey's Flea IEEE-1394 camera and an Hitachi video camera. They have smaller yet perceivable radial distortion. Figure-8 and figure-9 display the distortion correction results.

Figure 7: Distortion correction result for figure-5. (For display purpose we only show the center part of the image (though this would be no good for visual evaluation), because otherwise due to the large distortion the size of the corrected image would be too large to fit in with the paper size.)

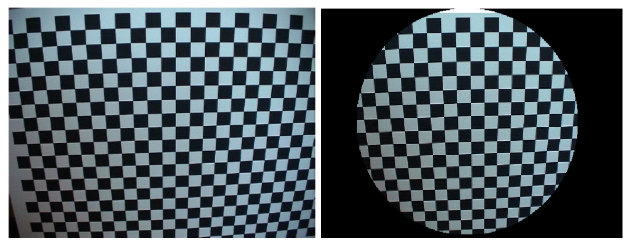

Figure 8: Radial correction result for the Flea camera. Left: input image; Right: output result. (only central-part is showed.)

8 Closure

In this paper, we show that the estimating of large lens radial distortion can be effectively decoupled from the estimating of other intrinsic and extrinsic camera parameters. Therefore, we are allowed to find the distortion parameters simply from two distorted images without bothering the fundamental (or essential) matrix. This increases the reliability of the algorithm, and provides better understanding of lens distortion. It also saves many unnecessary computations, and the obtained estimation is accurate enough for many practical applications.

The solution techniques we proposed here are basically

Figure 9: Radial correction result for the Hitachi video camera. Left: input image; Right: output result. (only central-part is showed.)

non-iterative. No initial estimation and no local minima, is needed or encountered. In order to solve the resulting simultaneous and hardly-consistent equation systems, we introduce a voting scheme. We discard the popular RANSAC idea, because it is quite inefficient in the problem context for two reasons: (1) it lacks an efficient way of doing inlier/outlier test; (2) noise significantly distorts the (minimal) basic equation itself. Our voting scheme gives reliable and robust results with respect to noise and outliers. The proposed algorithm is easy to implement. Our algorithm is best suitable for few-parameter case (say between 1-4). When the number of parameters is getting larger, potential instability may happen, as we deal with high degree polynomials, and the efficiency of KDE may get worse with too many parameters.

Currently we assume that the radial distortion model is algebraic. However, by some effort our method could be extended to some transcendental models. This will be our future work. We still need a better way to quantitatively evaluate the distortion-removal results for real image (e.g, measure the re-projection error.) Another practical issue is how to automatically match feature points in images under severe distortion.

Acknowledgments

National ICT Australia is funded through the Australian Governments Backing Australia's Ability Initiative, in part through the ARC. The real images used in experiments are courtesy of RH and S.Kang, token at MSR.

References

[1] T.Clarke,J.Fryer and W.Wang, The principal point for CCD cameras, Photogrammetric Record, 16(92):293-312,1998.

[2] Z.Zhang, On the Epipolar Geometry Between Two Images with Lens Distortion, In proc. ICPR-96, 1996.

[3] Z.Zhang, A Flexible New Technique for Camera Calibration, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.22, No.11, 1330C1334, November 2000.

[4] A. Fitzgibbon, Simultaneous Linear Estimation of Multipleview Geometry and Lens Distortion, in IEEE-CVPR-2001, Haway, December 2001.

[5] S.Thirthala and M.Pollefeys, The radial trifocal tensor: A tool for calibrating the radial distortion of wide-angle cameras, In Proc CVPR-2005, June 2005.

[6] D.Claus and A. Fitzgibbon, A rational function lens distortion model for general cameras, In Proc. CVPR-2005, June 2005.

[7] B.Micusik and T.Pajdla, Estimation of omnidirectional camera model from epipolar geometry, in Proc. CVPR-2003, 2003.

[8] R.Hartley, T.Saxena, The cubic rational polynomial camera model, In Proc. ARPA IU-Workshop'97, 1997.

[9] R.Hartley, S.B.Kang, Parameter-free Radial distortion correction with centre of distortion estimation, In Proc. ICCV-2005, Oct, 2005.

[10] R.Hartley,A.Zisserman, Multiview Geometry in Computer Vision,2nd Edition, Cambridge University Press, 2004.

[11] J.Philip, A Non-Iterative Algorithm for Determining all Essential Matrices Corresponding to Five Point Pairs, Photogrammetric Record, 15(88):589-599, 1996.

[12] D.Nister, An efficient five-point relative orentation algorithm, in proc. CVPR-2003, 2003.

[13] G.Stein, lens distortion calibration using point correspondences, in Proc. CVPR-1997, 1997.

[14] X. Ying, Z. Hu, Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model? In proc ECCV-2004,Prague, Czech, 2004.

[15] C. Geyer, K. Daniilidis, A Unifying Theory for Central Panoramic Systems and Practical Applications. in Proc. ECCV-2000,445-461, 2000.

[16] P. Sturm, A Method for 3D Reconstruction of Piecewise Planar Objects from Single Panoramic Images, OMNIVIS-2000, CVPR-2000 Workshop on Omnidirectional Vision, June 2000.

[17] J.Barreto and K. Daniilidis, Wide Area Multiple Camera Calibration and Estimation of Radial Distortion, Omnivis-2004, ECCV-2004 workshop, 2004.

[18] R.Swaminathan and S.K. Nayar,A Perspective on Distortions, In proc CVPR-2003, June 2003.

[19] S.Baker and S.K.Nayar, A theory of catadioptric image formation, In proc. ICCV-1998, pp 35-42, 1998.

[20] M.Grossberg and S.K.Nayar, A General Imaging Model and a Method for Finding its Parameters, in Proc ICCV-2001,Vancouver,Canada, July 2001.

[21] F.Devernay and O.Faugeraus, Striaght lines have to be straight, MVA, 13(1), 14-24,2001.

[22] S.B.Kang, Radial distortion snake,MVA, vol-12, 2000.

[23] T.Svoboda, T.Pajdla,H.Hlavac. Epipolar geometry for panoramic cameras, ECCV-1998,1998.

[24] J.Weng, P.Cohen, M.Herniou, Camera calibration with distortion models and accuracy evaluation,PAMI, 14(10),965-980,1992.