Real-time solution to the absolute pose problem with unknown radial distortion and focal length

Zuzana Kukelova
Czech Technical University, Faculty of Electrical Engineering,
Karlovo namesti 13, Prague, Czech Republic
kukelova@cmp.felk.cvut.cz

Martin Bujnak
Capturing Reality s.r.o.,
Bratislava, Slovakia
martin@capturingreality.com

Tomas Pajdla
Czech Technical University, Faculty of Electrical Engineering,
Karlovo namesti 13, Prague, Czech Republic
pajdla@cmp.felk.cvut.cz

Abstract

The problem of determining the absolute position and orientation of a camera from a set of 2D-to-3D point correspondences is one of the most important problems in computer vision with a broad range of applications. In this paper we present a new solution to the absolute pose problem for camera with unknown radial distortion and unknown focal length from five 2D-to-3D point correspondences. Our new solver is numerically more stable, more accurate, and significantly faster than the existing state-of-the-art minimal four point absolute pose solvers for this problem. Moreover, our solver results in less solutions and can handle larger radial distortions. The new solver is straightforward and uses only simple concepts from linear algebra. Therefore it is simpler than the state-of-the-art Gröbner basis solvers. We compare our new solver with the existing state-of-the-art solvers and show its usefulness on synthetic and real datasets. [1]

1. Introduction

Solving absolute camera pose, which means determining the position, orientation and possibly intrinsic camera parameters from n 2D-to-3D point correspondences, is known as the Perspective-n-Point (PnP) problem. The PnP problem is one of the oldest problems in computer vision [10] which creates a very basic element of many computer vision applications [22, 16, 15, 18]. Camera localization, structure from motion, scene reconstruction, object localization, tracking and recognition are just a few examples of such applications.

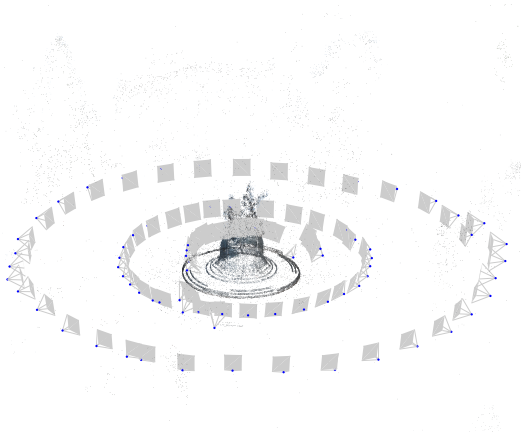

Figure 1. 3D reconstruction obtained using the new solver for the absolute pose problem for camera with unknown radial distortion and unknown focal length without further bundle adjustment.

The absolute pose problem for a fully calibrated camera was intensively studied in the past and many solutions were already developed. The problem itself can be formulated as a simple system of polynomial equations and solved in a closed form. Researchers in the past focused on how to formulate this problem, searched for different solutions, compared numerical stability, speed, or studied how to calculate the camera pose from more than three 2D-to-3D point correspondences [8, 19, 20, 21, 3, 24, 25].

Recently, a number of solutions to the absolute pose problems for cameras without complete internal calibration [2, 23, 4, 12, 5] and for cameras where some additional information about the scene is known [14] has been published. The reason for this is that in many applications

[1]This work has been supported by PRoViDE FP7-SPACE-2012-312377 and by De-Montes FP7-SME-2011-285839.

1550-5499/13 $31.00 © 2013 IEEE
DOI 10.1109/ICCV.2013.350

we do not have calibrated cameras, however, we can make some assumptions about these cameras or about the scene. For example, for modern cameras, we can usually assume square pixels and the principal point in the center of the image. On the other hand, consumer cameras have some non-negligible radial distortion and variable focal length.

The problem of estimating absolute pose of a camera together with its focal length for image points without radial distortion was first solved in [2], but only for planar scenes. The first solution to this focal length problem, which works for non-planar scenes, was presented by Triggs in [23]. The solution is based on multivariate resultants and works for non-planar scenes but fails for the planar ones.

An efficient solution to the absolute pose problem for camera with unknown focal length working for both planar and non-planar scenes was proposed recently in [4]. This solution is based on Euclidean rigidity constraint and for four 2D-to-3D point correspondences results in a system of four polynomial equations in four unknowns, which are solved using the Gröbner basis method [7]. The final solver is efficient and useful, however it may produce unsatisfactory results for cameras with radial distortion.

In [12] authors included the radial distortion to the problem with unknown focal length and proposed a method for solving absolute pose problem for a camera with radial distortion and unknown focal length from four 2D-to-3D point correspondences based on Gröbner bases. In experiments, authors show that in many real applications the consideration of radial distortion brings a significant improvement. The solution [12] uses one parameter division model for the radial distortion [9] and quaternions to parametrize rotations, and it results in five equations in five unknowns. These equations are quite complex and the Gröbner basis method [7] results in a large solver (a 1134 x 720 matrix) with 24 solutions. The final solver runs about $70ms$. Therefore, this solver is not practical in real-time applications.

A more practical solution to the absolute pose problem for a camera with unknown radial distortion and unknown focal length from four 2D-to-3D point correspondences was proposed in [5]. By decomposing the problem into a non-planar and planar case a much simpler and efficient solvers than in [12] were obtained. Both planar and non-planar solvers are again based on the Gröbner basis method for solving systems of polynomial equations [7]. The solution to the non-planar case requires to perform Gauss-Jordan (G-J) elimination of a 136×152 matrix and the eigenvalue computation of a 16×16 matrix. The planar solver requires G-J elimination of a 12×18 matrix. The proposed non-planar solver returns 16 and the planar solver 6 solutions. These two solvers can be joined to a single general solver. The joined solver gives comparable or better results than the general solver [12] for most scenes, including the near-planar ones. Its running time is about $700\mu s$ for original Matlab implementation available on [1] and $260\mu s$ for our fast C++ implementation based on methods from [6]. This solver is much faster than the solver [12], however it is still much more complicated than the solvers to the calibrated absolute pose problem [8] or the problem with unknown focal length [4]. Moreover, the solver [5] is quite complicated to implement without the special Gröbner basis software [13].

In this paper we propose a new solutions to the important absolute pose problem for camera with unknown radial distortion and unknown focal length. In this solution we extend the solution proposed in [5]. Thanks to having five 2D-to-3D point correspondences we obtain simpler and significantly faster solver that results in less solutions and can handle larger radial distortions than the solvers presented in [5, 12]. The new solver can work with up to three parameter division model for radial distortion [9] and provides up to four real solutions. It is a minimal solver for three parameter division model, i.e. it uses the minimal number of correspondences needed to solve this problem.

By evaluating our new solver on synthetic and real data, we show that it is numerically more stable and more accurate than the state-of-the-art solvers [12, 5]. The running time of the new solver is $2\mu s$. Therefore this solver is very useful in real applications where consideration of radial distortion may bring a significant improvement over the absolute pose problems without distortion [8, 4].

2. Problem Formulation

Let us assume the standard pinhole camera model [11]. In this model the image projection $\mathbf{u}_i$ of a 3D point $\mathbf{X}_i$ can be written as

$$\alpha_i \, \mathbf{u}_i = \mathtt{P} \, \mathbf{X}_i, \tag{1}$$

where $\mathtt{P}$ is a 3×4 projection matrix, α_i is an unknown scalar value and points $\mathbf{u}_i = [u_i, v_i, 1]^\top$ and $\mathbf{X}_i = [X_i, Y_i, Z_i, 1]^\top$ are represented by their homogeneous coordinates.

The projection matrix $\mathtt{P}$ can be written as

$$\mathtt{P} = \mathtt{K} \, [\mathtt{R} \,|\, \mathbf{t}], \tag{2}$$

where $\mathtt{R} = [r_{ij}]_{i,j=1}^{3}$ is a 3×3 rotation matrix, $\mathbf{t} = [t_x, t_y, t_z]^\top$ is the camera position vector and $\mathtt{K}$ is the 3×3 calibration matrix of the camera.

We assume that the only unknown parameter from the calibration matrix $\mathtt{K}$ is the focal length. Therefore, the calibration matrix $\mathtt{K}$ has the form $diag\,[f, f, 1]$. Since the projection matrix is given only up to scale we can equivalently write $\mathtt{K} = diag\,[1, 1, w]$ for $w = \frac{1}{f}$ and the projection matrix (2) has the form

$$\mathtt{P} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ wr_{31} & wr_{32} & wr_{33} & wt_z \end{bmatrix}. \tag{3}$$

The projection equation (1) holds for the pinhole camera model with no radial distortion. However, in real situations we measure image points $\hat{\mathbf{u}}_i$ affected by some radial distortion. To use measured distorted image points $\hat{\mathbf{u}}_i$ in (1) we need to "undistort" them. Therefore, we obtain

$$\alpha_i \, f_u(\hat{\mathbf{u}}_i) = \mathtt{P} \, \mathbf{X}_i, \tag{4}$$

where f_u is a function that undistort an image point and $\alpha_i \in \mathbb{R}$ is an unknown scalar value.

The previous solvers [12, 5] to the absolute pose problem for camera with unknown radial distortion used equation (4) and the one-parameter division model for distortion [9]. In this model $f_u(\hat{\mathbf{u}}_i) = \left[\hat{u}_i, \hat{v}_i, 1 + k(\hat{u}_i^2 + \hat{v}_i^2)\right]^\top$, where $\hat{\mathbf{u}}_i = [\hat{u}_i, \hat{v}_i, 1]^\top$ are the homogeneous coordinates of the distorted image point $\hat{\mathbf{u}}_i$ and k is a distortion parameter.

In [17], it was shown that the two-parameter division model may give in some applications better results than the one-parameter division model.

In our new five point absolute pose solver we can handle up to three parameters for radial distortion without increasing the complexity of the solver. We will consider up to three parameter division model of the form

$$f_u(\hat{\mathbf{u}}_i) = \left[\hat{u}_i, \hat{v}_i, 1 + k_1 \hat{r}_i^2 + k_2 \hat{r}_i^4 + k_3 \hat{r}_i^6\right]^\top \tag{5}$$

where $\hat{r}_i^2 = \hat{u}_i^2 + \hat{v}_i^2$ and $k_1, k_2, k_3 \in \mathbb{R}$ are distortion parameters.

Now we describe our new solver for the problem of estimating absolute pose of a camera with unknown radial distortion and unknown focal length from five 2D-to-3D point correspondences.

3. Absolute pose for a camera with radial distortion and unknown focal length

Our new solver follows the formulation used in [5], but the new solver, unlike the solver from [5], works for non-planar as well as for planar scenes. The difference is in the solving method. While the non-planar P4Pfr solver from [5] assumes regularity of some matrices and therefore fails for planar scenes, our new P5Pfr solver does not make such assumptions and works for both planar and non-planar scenes.

For planar scenes, a slightly simpler solver can be created but this is not necessary, since the new P5Pfr solver is already very simple, fast and efficient.

In our new solver we start by eliminating the scalar values α_i from the projection equation (4). We do this by multiplying (4) by the skew symmetric matrix $[f_u(\hat{\mathbf{u}}_i)]_\times$. Since $[f_u(\hat{\mathbf{u}}_i)]_\times \, f_u(\hat{\mathbf{u}}_i) = 0$ we obtain the matrix equation

$$\begin{bmatrix} 0 & -\hat{w}_i & \hat{v}_i \\ \hat{w}_i & 0 & -\hat{u}_i \\ -\hat{v}_i & \hat{u}_i & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = 0, \tag{6}$$

where $\hat{w}_i = 1 + k_1 \hat{r}_i^2 + k_2 \hat{r}_i^4 + k_3 \hat{r}_i^6$, $\mathbf{X}_i = [X_i, Y_i, Z_i, 1]^\top$ and p_{ij} is the element from the i^{th} row and j^{th} column of the projection matrix $\mathtt{P}$.

This matrix equation gives three polynomial equations from which only two are linearly independent. This is caused by the fact that the skew symmetric matrix $[f_u(\hat{\mathbf{u}}_i)]_\times$ has rank two.

Let's now consider the equation corresponding to the third row of the matrix equation (6). This equation can be written as

$$\begin{aligned} -\hat{v}_i \, (p_{11} \, X_i + p_{12} \, Y_i + p_{13} \, Z_i + p_{14}) + \\ +\hat{u}_i \, (p_{21} \, X_i + p_{22} \, Y_i + p_{23} \, Z_i + p_{24}) = 0. \end{aligned} \tag{7}$$

This is a homogeneous linear equation in eight unknowns $p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}$ and p_{24}. Since we have five 2D-to-3D point correspondences, we have five equations of the form (7). These five equations can be rewritten in the matrix form

$$\mathtt{M} \, \mathbf{v} = 0, \tag{8}$$

where $\mathtt{M}$ is a 5×8 coefficient matrix and $\mathbf{v} = [p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}]^\top$ is a 8×1 vector of unknowns. Therefore we can write our eight unknowns in $\mathbf{v}$ as a linear combination of the three null space basis vectors $\mathbf{n}_i$ of the matrix $\mathtt{M}$

$$\mathbf{v} = \sum_{i=1}^{3} \gamma_i \, \mathbf{n}_i, \tag{9}$$

where γ_i are new unknowns from which one, e.g. γ_3, can be set to one.

In this way we obtain a parametrization of the first two rows of the projection matrix $\mathtt{P}$ with two unknowns γ_1 and γ_2.

To find the solutions for γ_1 and γ_2 we use constraints that the first two rows of the 3×3 submatrix of the projection matrix $\mathtt{P}$ are mutually perpendicular and have the same norm, i.e. the constraints

$$p_{11} p_{21} + p_{12} p_{22} + p_{13} p_{23} = 0, \tag{10}$$

$$p_{11}^2 + p_{12}^2 + p_{13}^2 - p_{21}^2 - p_{22}^2 - p_{23}^2 = 0. \tag{11}$$

These constraints follow from the fact that the 3×3 submatrix of the projection matrix $\mathtt{P}$ has the form $\mathtt{K}\,\mathtt{R}$, where $\mathtt{R}$ is a rotation matrix. In this way we obtain two quadratic equations in two unknowns γ_1 and γ_2. These two equations can be written as quadratic equations in only one unknown γ_1 and with polynomial coefficients in γ_2. Then, the Sylvester resultant matrix [7] of these two polynomials can be used to efficiently solve them. The determinant of this 4×4 Sylvester matrix [7] results in a fourth degree polynomial in γ_2 which can be solved in a closed form. In this

way we obtain up to four real solutions to the first two rows of the projection matrix P.

Next we use the constraints that the third row of the 3×3 submatrix of the projection matrix P is perpendicular to the first two rows of this submatrix, i.e. the constraints

$$p_{31}p_{11} + p_{32}p_{12} + p_{33}p_{13} = 0, \quad (12)$$

$$p_{31}p_{21} + p_{32}p_{22} + p_{33}p_{23} = 0. \quad (13)$$

Using these two linear homogeneous equations we can parametrize p_{31}, p_{32} and p_{33} with only one unknown. Let's denote it by δ.

To find the solutions to the unknown radial distortion parameters and to the third row of the projection matrix P, i.e. solutions to δ and p_{34}, we use one of the two unused equations from the projection equation (6). To guarantee numerical stability, we use the equation corresponding to the first row of (6) when $|\hat{u}_i| < \epsilon$, for some small ϵ, and the equation corresponding to the second row when $|\hat{v}_i| < \epsilon$. In all the remaining situations, which are the most common, we select arbitrarily from these two equations, e.g., the equation corresponding to the second row. Then we get

$$(1 + k_1\hat{r}_i^2 + k_2\hat{r}_i^4 + k_3\hat{r}_i^6)(p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}) - \quad (14)$$
$$-\hat{u}_i\,(p_{31}\,X_i + p_{32}\,Y_i + p_{33}\,Z_i + p_{34}) = 0.$$

Since p_{11}, p_{12}, p_{13} and p_{14} are already known and p_{31}, p_{32} and p_{33} are parameterized with δ, equation (14) is a (non-homogeneous) linear equation in five unknowns δ, p_{34}, k_1, k_2 and k_3. We again have five linear equations of the form (14) from which we can easily obtain solutions to these five unknowns.

In many applications, it is better to use only one or two parameter division model not to overfit data. In such a case we obtain an overconstrained system of five linear equations in three or four unknowns, which we can solve using the SVD decomposition.

Finally, we use the constraint that the squared norm of the first row of the left 3×3 submatrix of the projection matrix P multiplied by w^2 is equal to the squared norm of the third row of this submatrix

$$w^2\,p_{11}^2 + w^2\,p_{12}^2 + w^2\,p_{13}^2 - p_{31}^2 - p_{32}^2 - p_{33}^2 = 0, \quad (15)$$

from which we extract solutions to the unknown focal length $f = \frac{1}{w}$.

4. Experiments

We have tested our new P5Pfr solver on synthetic data (with various radial distortions, focal lengths, outliers and levels of noise) and on real datasets, evaluated its speed, stability and precision and compared it with the state-of-the-art four point P4Pfr solvers to the absolute pose problem for camera with unknown focal length and radial distortion [12, 5], the P3P algorithm for calibrated camera presented in [8] and the P4Pf algorithm for camera with unknown focal length [4]

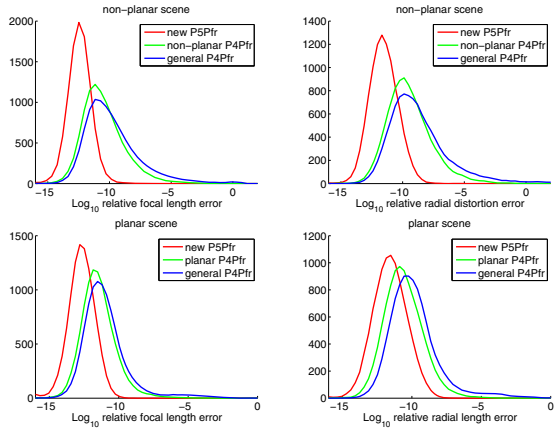

Figure 2. Log_{10} of the relative error of the focal length f and log_{10} of the relative error of the radial distortion parameter k_1 obtained by selecting the real root closest to the ground truth value for non-planar scenes (Top) and planar scenes (Bottom).

4.1. Synthetic data

First we studied the new solver on synthetically generated ground-truth 3D scenes. These scenes were generated randomly using 3D points randomly distributed in a cube or on a plane depending on the testing configuration. Each 3D point was projected by a camera with random or fixed focal length. The camera orientation and position were selected randomly but looking on the scene. Then the radial distortion using one or two-parameter division model was added to all image points to generate noiseless distorted points. Finally, Gaussian noise with standard deviation σ was added to the distorted image points assuming a 1000×1000 pixel image.

4.1.1 Numerical stability

In the first experiment we have studied the behavior of the new solver on noise free data to check its numerical stability and compared the results with the numerical stability of the state-of-the art P4Pfr solvers [12, 5].

In this experiment we have generated 10000 scenes with 3D points randomly distributed in a cube or on a plane and cameras with random feasible position and orientation. To be able to compare our new solver with the state-of-the-art solvers [12, 5] we have used the one-parameter division model for radial distortion in our solver ($k_2 = 0$ and $k_3 = 0$ in (5)). The radial distortion parameter k_1 was randomly drawn from the interval $k_1 \in [-0.45, 0]$ and the focal length from the interval $f \in [0.5, 2.5]$ in this experiment.

Figure 2 shows the results of our new P5Pfr solver on non-planar (Top) and on planar scenes (Bottom). In both cases we compare our new solver (Red) with the general solver from [12] (Blue) and the non-planar and planar solvers from [5] (Green).

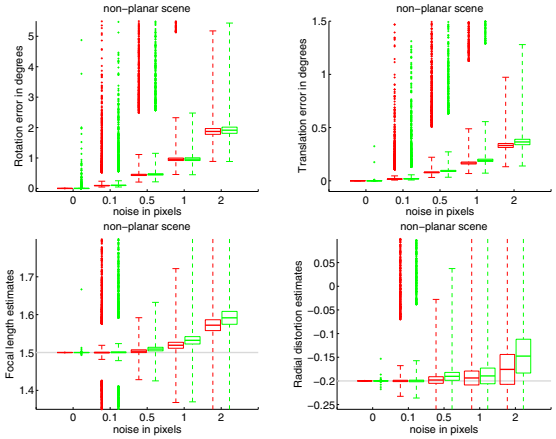

Figure 3. Error of rotation (Top left), error of translation (Top right), focal length estimates (Bottom left) and radial distortion estimates (Bottom right) in the presence of noise for the new P5Pfr solver (Red) and the P4Pfr non-planar solver [5] (Green).

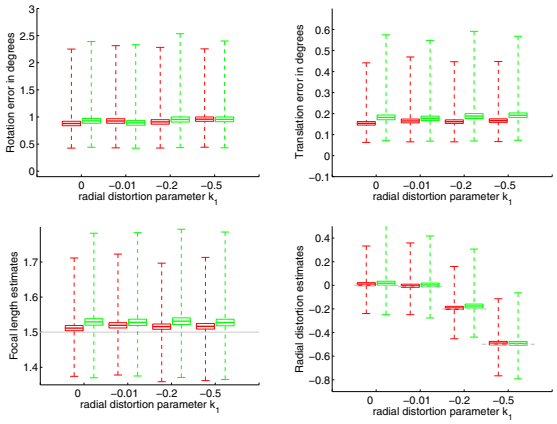

Figure 4. Error of rotation (Top left), error of translation (Top right), focal length estimates (Bottom left) and radial distortion estimates (Bottom right) in the presence of $1px$ noise and different radial distortions for the new P5Pfr solver (Red) and the P4Pfr non-planar solver [5] (Green).

Left plots in Figure 2 show the log_{10} of the relative error of the focal length f obtained by selecting the real root closest to the ground truth value. Right plots in Figure 2 show the log_{10} of the relative error of the radial distortion parameter k_1.

The new P5Pfr solver is slightly numerically more stable than the state-of-th-art P4Pfr solvers [12, 5] for both planar and non-planar scenes. The important fact is that the new solver gives fewer large errors than [12, 5]. Our new solver did not return error greater than 10^{-5} in this noise free experiment. The general P4Pfr solver [12] "failed", i.e. returned error greater than 10^{-5}, in about 3.5% for non-planar scenes and in about 1.3% for planar scenes. The non-planar P4Pfr solver [5] "failed" in about 1.6% and the planar P4Pfr solver in about 0.13%.

The similar numerical stability of our new solver is obtained also for the two and three parameter division model.

4.1.2 Noise test

In the next experiment we have studied the accuracy of our new solver in the presence of noise added to image points. Figure 3 shows the results for non-planar 3D scene and different levels of noise added to image projections. In this case the ground truth focal length was $f_{gt} = 1.5$ and the radial distortion $k_{1_{gt}} = -0.2$. Results for planar scenes were similar and are not shown here. Since both P4Pfr solvers [12, 5] return very similar results, we compared in this experiment our new solver only with the more efficient non-planar solver from [5].

For each noise level, from 0 to 2 pixels, 5000 estimates for random non-planar scenes and random feasible camera positions were made. Results in Figure 3 are represented by the Matlab boxplot function which shows values 25% to 75% quantile as a box with horizontal line at median. The crosses show data beyond 1.5 times the interquartile range.

In this case the rotation error (Top left) was measured as the rotation angle in the angle-axis representation of the relative rotation $\mathtt{RR}_{gt}^{-1}$ and the translation error (Top right) as the angle between the ground-truth and the estimated translation vector. Plots in Figure 3 (Bottom) show directly the estimates of the focal length (Left) and the radial distortion parameter k_1 (Right).

It can be seen that the new solver gives more accurate estimates than the state-of-the-art solver [5] for all rotation, translation, focal length and radial distortion. The new solver is performing very well even at two pixel noise level.

4.1.3 Different radial distortions

In this experiment we have tested the behavior of the new P5Pfr solver for increasing radial distortion. Figure 4 shows the results for general non-planar 3D scene, camera with focal length $f_{gt} = 1.5$, image noise $1px$ and increasing radial distortion parameter k_1. We compare our new solver (Red) with the non-planar P4Pfr solver [5] (Green).

For each radial distortion, from $k_{1_{gt}} = 0$ (no radial distortion) to $k_{1_{gt}} = -0.5$ (large radial distortion), 5000 estimates for random non-planar scenes and random feasible camera positions were made. Results in Figure 4 are represented by the Matlab boxplot function and shows the rotation error (Top left), translation error (Top right), focal length estimates (Bottom left) and radial distortion estimates (Bottom right).

We can see that the accuracy of the new solver does not depend on the radial distortion. The new solver gives accurate and stable results for all radial distortions including

scenes without radial distortion and it slightly outperforms the non-planar P4Pfr solver [5].

4.2. Computational effort

The most significant improvement of our new solver over the state-of-the-art solvers [12, 5] is in the speedup. The C++ implementation of the new P5Pfr solver runs about $2\mu s$ and is thus about $130\times$ faster than the C++ implementation of the non-planar P4Pfr solver [5] which runs about $260\mu s$. The speed-up over the general P4Pfr solver [12] is even much bigger.

The new solver requires to compute a null space of a 5×8 matrix, to find solutions to a fourth degree polynomial and to compute inverse of a 3×3, 4×4 or 5×5 matrix, depending on how many radial distortion parameters (one, two, or three) are considered.

For comparison, the general P4Pfr solver [12] requires to perform LU decomposition of a 1134×720 matrix, QR decomposition of a 56×56 matrix and eigenvalue computations of a 24×24 matrix. The non-planar P4Pfr solver [5] requires to perform G-J elimination of a 136×152 matrix and eigenvalue computations of a 16×16 matrix.

The mean number of feasible real solutions returned by the new solver for non-planar scenes is 1.7. For planar scenes our new solver returns one solution with multiplicity two. The mean number of real solutions returned by the non-planar P4Pfr solver [5] is 3.7 and by the planar P4Pfr solver [5] 4.6.

Our new solver uses one more point than the previous minimal four point solutions to the absolute pose problem with radial distortion [12, 5]. Therefore, theoretically, the new solver requires more samples in RANSAC-based algorithms [8] for reaching the same probability of finding an uncontaminated sample. However, the new solver is about $130\times$ faster than the fastest four point radial distortion absolute pose solver [5] and it returns also less solutions than [5], which need to be evaluated. Therefore, the running time of a RANSAC-based algorithm [8] for the new P5Pfr solver is significantly smaller than the running time of a RANSAC-based algorithm for the P4Pfr solver [5] for all reasonable outlier contaminations and scenes.

In fact, even for a very simple implementation of the "vanilla" RANSAC algorithm [8] with 1000 RANSAC cycles and 2000 correspondences between 2D and 3D space, it takes only $50ms$ to estimate the camera pose, radial distortion and focal length using the new P5Pfr solver. The major fraction of this time is used for pose verification. Therefore, the new algorithm can be very efficiently used in real-time applications.

The same RANSAC algorithm [8] with our fast C++ implementation of the P4Pfr solver [5] runs around $300ms$ on the same data. Major fraction of this time is used for pose calculation.

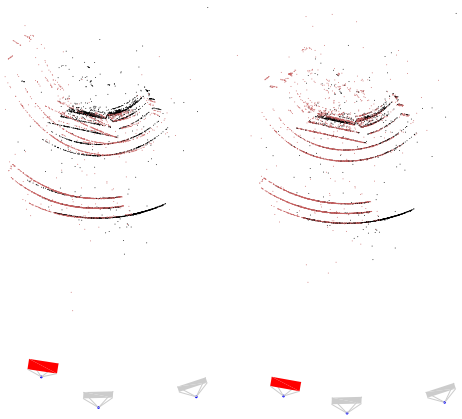

Figure 5. Camera registered using the P3P algorithm [8] and focal length extracted from $EXIF$ (Left) and camera estimated using the new P5Pfr solver (Right). The black point cloud was calculated using gray cameras only. The red point cloud was created by triangulating 2D matches between red and gray cameras. Notice a strong misalignment in the P3P result (Left) caused by a small radial distortion present in the red camera.

The comparison of the total times of model computation and 2000 tentative matches verification in RANSAC [8] loop and different outlier contaminations is shown in Table 1. The total times of solvers computation together with the verification, 0.3ms/sample for the P4Pfr solver [5] and 0.05ms/sample for the new P5Pfr solver, were obtained by measuring the mean time of 1000 samples in real experiments.

From Table 1 it can be seen that the new P5Pfr solver does 460515 samples while the P4Pfr solver [5] does 46049 at 90% outlier contamination. However, $0.921s$ of the new P5Pfr solver spent in the model computation is $13\times$ less than $11.973s$ of the P4Pfr [5]. The total time, including the verification of 2000 tentative matches, of the new P5Pfr solver is shorter than the P4Pfr solver: $5\times$ at 10%, $3\times$ at 50%, and $1.2\times$ at 80% outlier contamination.

4.3. Real data experiment

To evaluate our new solver on a real data, we created a simple structure from motion pipeline. The pipeline detects image features and match them like in [22]. We manually picked an image pair (initial seed) with a visually low radial distortion and calculated the relative pose of its cameras using [6]. We used a fixed number of iterations and a simple RANSAC [8] algorithm. To calibrate the camera pair, we used the focal lengths extracted from image $EXIF$s. Then, we triangulated image feature correspondences which were accepted as inliers in the relative pose RANSAC. This way we obtained a 3D point cloud. Using the correspondences between images we created a set of 2D-3D correspondences for images which were not registered yet. Next, we used our

Outliers	10%	30%	50%	70%	80%	90%
P4Pfr #samples	5	17	72	566	2876	46049
P5Pfr #samples	6	26	146	1893	14389	460515
P4Pfr solver time [s]	0.00130	0.00442	0.0187	0.1472	0.7478	11.973
P5Pfr solver time [s]	0.00001	0.00005	0.0003	0.0038	0.0288	0.921
P4Pfr total time [s]	0.00150	0.00510	0.0216	0.1698	0.8628	13.814
P5Pfr total time [s]	0.00030	0.00130	0.0073	0.0947	0.7195	23.026
P5Pfr total time / P4Pfr total time	0.200	0.255	0.338	0.557	0.834	1.667

Table 1. The comparison of the total times of model computation and 2000 tentative matches verification in RANSAC [8] loop and different outlier contaminations for the P4Pfr solver [5] and the new P5Pfr solver.

new P5Pfr absolute pose solver to register these images one by one. To achieve that, we were executing our new solver 1000 times inside the RANSAC [8] loop and this way we obtained the pose, focal length and distortion coefficients of each camera. Next we undistorted its measurements and triangulated new 3D points using 2D matches between already registered images. Then we created a new set of 2D-3D correspondences to register remaining images. This way we incrementally registered all images. The result of reconstruction is shown in Figure 1.

In fact, disregarding some implementation details, this is a traditional incremental structure from motion pipeline. Typically the calibrated P3P algorithm [8] with focal length extracted from $EXIF$ or the DLT algorithm [11] is used in such a pipeline. However, this can produce very inaccurate results even if camera lens distortion is almost invisible. This happens not only for wide field of view cameras but also for widely used standard consumer cameras or smart phones and even at $27mm$ focal length. Figure 5 (Left) shows such a failure for Nikon D90 camera with 18-$105mm$ lens. The left plot shows the result, using the above pipeline with the standard P3P algorithm, which assumes no radial distortion. The right plot of Figure 5 shows the result for the new P5Pfr algorithm where it is visible that the seed and newly triangulated point clouds smoothly overlap. In both cases, initial seed camera pair (the middle and the right camera) were captured at $35mm$ focal length with respect to $35mm$ film size. The distorted image (left camera) was captured using the same Nikon camera at $27mm$.

To avoid registration failures, typically, bundle adjustment [11] initialized with an absolute pose estimated in RANSAC is used. Radial distortion and focal length are optimized inside the bundle adjustment as well. However, calculating radial distortion and focal length inside RANSAC allows finding much more accurate initialization and also finding more inlier correspondences which are crucial for successful optimization too. The speed at which RANSAC gains inliers is shown in Figure 6. This results were obtained on the real data from the previous experiment, i.e. the statue from Figure 1. There was approximately 20% of outliers since every 2d-3d point correspondence was at least

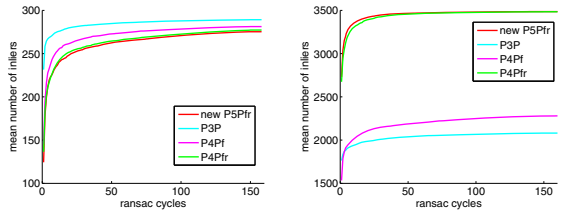

Figure 6. The mean number of inliers (y-axis) obtained inside of the RANSAC as a function of the number of iterations (x-axis). Results for 500 calls of RANSAC for an image with low distortion (Left) and for an image with higher distortion (Right).

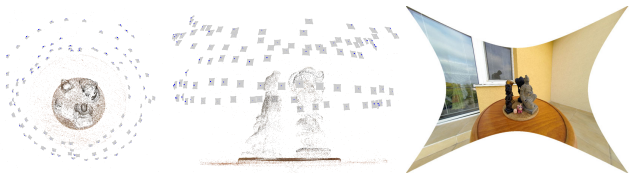

Figure 7. A scene reconstructed using a mixture of images with small and significant radial distortion (Left and Middle). An image undistorted using parameters calculated by the new P5Pfr solver (Right).

three-view consistent. From Figure 6 it can be seen that the speed is very slow for solvers which does not calculate radial distortion. Also such solvers did not gain even half of inliers.

Looser thresholds can be used to gain more correspondences faster but then we need a better tentative correspondences, otherwise we easily include outliers into the inlier set. On the other hand, calculating a stronger model inside the RANSAC allows using tighter thresholds for outlier classification and hence reducing possible contamination.

For reconstructing scene in Figure 1 we did neither use bundle adjustment nor any other local optimization method.

In our last experiment we tested the new P5Pfr camera pose solver on images with significant radial distortion. First, we created an initial reconstruction using a camera with a small radial distortion. The camera captured images in a single loop around an testing object. We used the same reconstruction method as in the previous real experiment.

We optimized the computed camera poses using bundle adjustment. Next, we captured another set of images around the object using a camera with significant radial distortion - GoPro camera with 170 degree fish eye lens. We registered these images to the scene using the new P5Pfr solver and we triangulated new 3D points. We did not further optimized the estimated camera poses and calibrations. The results are displayed in Figure 7.

5. Conclusion

In this paper we have proposed a new efficient solution to the absolute pose problem for camera with unknown radial distortion and unknown focal length from five 2D-to-3D point correspondences. Thanks to having five correspondences, we have obtained a simpler and significantly faster solver that gives less solutions to test and can handle larger radial distortions than the state-of-the-art solvers presented in [5, 12]. The new solver is about $130\times$ faster than the fastest absolute pose solver for camera with radial distortion [5]. By evaluating our new solver on synthetic and real data we have shown that it is numerically more stable and more accurate than the state-of-the-art solvers [12, 5].

The new solver is straightforward and uses only simple concepts from linear algebra. Therefore, it is simpler to implement than the Gröbner basis solvers [12, 5]. The running time of the new solver is $2\mu s$. This solver is very useful in real applications where consideration of radial distortion may bring a significant improvement over the absolute pose problems without distortion [8, 4] as we have shown in real experiments.

References

[1] http://cmp.felk.cvut.cz/minimal. 2

[2] M. A. Abidi and T. D. Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: algorithm and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):534–538, May 1995. 1, 2

[3] M.-A. Ameller, B. Triggs, and L. Quan. Camera Pose Revisited – New Linear Algorithms, 2000. Submitted to ECCV'00. 1

[4] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *CVPR'08*, 2008. 1, 2, 4, 8

[5] M. Bujnak, Z. Kukelova, and T. Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *ACCV'10*, pages 11–24, 2010. 1, 2, 3, 4, 5, 6, 7, 8

[6] M. Bujnak, Z. Kukelova, and T. Pajdla. Making Minimal Solvers Fast. In *CVPR'12*, 2012. 2, 6

[7] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer, 2nd edition, 2005. 2, 3

[8] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 1, 2, 4, 6, 7, 8

[9] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR'01*, volume 1, page 125, Los Alamitos, CA, USA, 2001. IEEE Computer Society. 2, 3

[10] J. A. Grunert. Das pothenot'sche problem, in erweiterter gestalt, nebst bemerkungen über seine anwendung in der. In *Archiv der Mathematik und Physik*, page 238248, 1841. 1

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004. 2, 7

[12] K. Josephson and M. Byröd. Pose estimation with radial distortion and unknown focal length. In *CVPR'09*, 2009. 1, 2, 3, 4, 5, 6, 8

[13] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *ECCV'08, Part III*, volume 5304 of *Lecture Notes in Computer Science*, 2008. 2

[14] Z. Kukelova, M. Bujnak, and T. Pajdla. Closed-form solutions to the minimal absolute pose problems with known vertical direction. In *ACCV'10*, 2, pages 216–229, 2010. 1

[15] B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV'07*, 2007. 1

[16] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *Proceedings of the 12th European conference on Computer Vision - Volume Part I*, ECCV'12, pages 15–29, Berlin, Heidelberg, 2012. Springer-Verlag. 1

[17] L. Ma, Y. Chen, and K. L. Moore. Rational radial distortion models of camera lenses with analytical solution for distortion correction. *I. J. Information Acquisition*, 1(2):135–147, 2004. 3

[18] Microsoft. Photosynth - http://www.photosynth.net. 1

[19] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *ICCV'07*, pages 1–8, October 2007. 1

[20] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, August 1999. 1

[21] G. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. In *ISSAC'03*, pages 215–223, 2003. 1

[22] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. 1, 6

[23] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *ICCV'99*, volume 1, page 278, 1999. 1, 2

[24] Y. Wu and Z. Hu. Pnp problem revisited. *Journal of Mathematical Imaging and Vision*, 24:131–141, 2006. 1

[25] L. Zhi and J. Tan. A complete linear 4-point algorithm for camera pose determination. *Academy of Mathematics and System Sciences, Academia Sinica*, 21:239–249, 2002. 1