Pose Estimation with Radial Distortion and Unknown Focal Length

Klas Josephson and Martin Byröd
{klasj, byrod}@maths.lth.se
Centre for Mathematical Sciences, Lund University, Lund, Sweden
http://www.maths.lth.se/vision

Abstract

This paper presents a solution to the problem of pose estimation in the presence of heavy radial distortion and a potentially large number of outliers. The main contribution is an algorithm that solves for radial distortion, focal length and camera pose using a minimal set of four point correspondences between 3D world points and image points. We use a RANSAC loop to find a set of inliers and an initial estimate for bundle adjustment. Unlike previous approaches where one starts out by assuming a linear projection model, our minimal solver allows us to handle large radial distortions already at the RANSAC stage. We demonstrate that with the inclusion of radial distortion in an early stage of the process, a broader variety of cameras can be handled than was previously possible. In the experiments, no calibration whatsoever is applied to the camera. Instead we assume square pixels, zero skew and centered principal point. Although these assumptions are not strictly true, we show that good results are still obtained and by that conclude that the proposed method is applicable to uncalibrated photographs.

1. Introduction

The ability to find the position and the direction in which a camera points is an old and challenging problem in computer vision. If an image based approach is chosen, as in this paper, the common way to solve the problem is to find correspondences between an image taken with a camera with unknown position and a three dimensional model. This method has for example been used in Photo tourism [25]. In this paper we choose to follow the same outline of the algorithm but add one extra component to the model, radial distortion. The enhancement with radial distortion makes it possible to use photos taken with fisheye lenses and other heavily distorted images, see Figure 1 for an example.

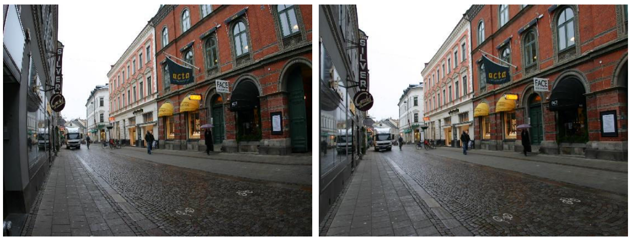

Figure 1. Left: An image taken with a fisheye lens. Right: The same image rectified when kernel voting is used to determine the radial distortion

The oldest papers on localization are from the time before the research field of computer vision existed. Already in 1841 Grunert [14] showed that there can be up to four real solutions to the problem of localization if inner calibration of the camera is known and there are three correspondences between the images and known three dimensional points. For an easier description of the problem and how to solve it, [15] is recommended.

If the inner calibration is unknown it is necessary to have six correspondences between the image and the 3D model. In that case a linear method to find the camera position exists [16]. This method usually gives poor results since digital cameras have square pixels and the principal point close to the center of the image. By not imposing these assumptions to the camera model, too many degrees of freedoms are used which makes the model unnecessarily unstable. These assumptions can however be incorporated and the problem is then to find the pose along with an unknown focal length. In 1995 Abidi and Chandra [1] presented a solution to this problem that worked on planar scenes. Four years later Triggs [28] gave a solution to the same problem that worked well on non-planar scenes. In the same paper he also presented a solution to the same problem but without any assumptions on the principal point of the camera. In 2008 the latest paper [4] on this problem was presented. In this paper Bujnak *et al.* presents a solution that works on both planar and non-planar data. In that solution Gröbner basis methods were used to solve the system of polynomial equations that arises in their solution. Gröbner bases were also mentioned in the paper by Triggs and to the authors knowledge this is the first paper in the computer vision community that uses Gröbner basis methods to solve a system of polynomial equations. This is also the method that will be used in this paper to solve the systems of polynomial equations arising in the problem.

The problem of pose with unknown focal length is not a true minimal case with four points, hence no exact solution can be found. In [4] the fact that the problem is over constrained is resolved by ignoring one equation in an early step of the solver and then using the last equation to verify which of the multiple solutions to use. An alternative method to find the focal length was presented by Josephson *et al.* [18]. In that paper a correspondence to another image replaced one of the correspondences to a three dimensional point. That method can also be used for the four points problem if one of the points is substituted by an arbitrary line through that point. In this work we instead choose to include radial distortion into the model. This adds one degree of freedom and hence the four points problem becomes minimal.

Minimal problems such as those described above and the one presented in this paper are in computer vision usually used as the key component in a RANSAC engine [10]. RANSAC is the most commonly used method to estimate camera pose from an image, also used in this paper, works as follows. Start by finding correspondences between the image and a model, this is usually done by finding interest points in the images and calculate descriptors of these, see [2, 23, 24]. In this paper SIFT is used. After that the RANSAC engine is used to find consistent correspondences, so called inliers. The inliers are in the end used in a local optimization initiated by the camera model given by the RANSAC engine.

The contribution of this paper is to use radial distortion already in the RANSAC step in the problem of absolute pose. Radial distortion was introduced to the computer vision community by Devernay and Faugeras [9] in 1995. But it was used long before that in the photogrammetry literature, *e.g.* [3]. In both these paper the so-called "plumb line model" is used. This model is probably the most used model, *e.g.* Photo tourism uses this model. However this model is not well suited for minimal problems. Instead we use the division model introduced by Fitzgibbon in 2001 [11]. In that paper he shows that this model is equally powerful as the plumb line model. Due to its simpler form, the division model has been used on several minimal problems [6, 17, 20].

Although the main focus of this paper is the pose estimation problem, the solver of the minimal solution can also be used to estimate the focal length and the radial distortion of a camera lens and by that un-distort an image. A method to find the radial distortion is to use kernel voting. In [22] Li and Hartley used it to find the radial distortion and in [21] Li used it to find the focal length.

2. The Camera Model

The basis of the camera model used in this paper is the standard pinhole camera model [16] where the projection equation is written,

$$\lambda \mathbf{x} = P\mathbf{X}. \tag{1}$$

Here, P is the so-called camera matrix of size 3×4. The camera matrix can be factorized as,

$$P = K[R \mid \mathbf{t}]. \tag{2}$$

In this factorization R is a rotation matrix and holds the information in which direction the camera is pointing and $\mathbf{t}$ gives information of camera position. K is the calibration matrix of the camera and compensates for the intrinsic setup of the camera. The K matrix can be written

$$K = \begin{bmatrix} f & s & p_x \\ 0 & \gamma f & p_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

In this matrix f represents the focal length of the camera. Further on represents s the skew, for most digital cameras this is zero and the aspect ratio of the pixels described by γ is very close to one. The principal point of these cameras given by (p_x, p_y) is also close to the center of the images. In the rest of the paper a principal point in the center of an image and square pixels with zero skew are assumed and it will be showed that in practice, these assumptions yield good results even though they are not strictly true.

In this paper the pinhole camera model is extended with radial distortion. The radial distortion is modeled by the division model introduced by Fitzgibbon [11]. The reason to choose this model is that it gives easier calculations than the plumb line model. The model transforms the distorted coordinates given by the pinhole camera model according to,

$$\mathbf{p}_u = \mathbf{p}_d/(1 + \mu r_d^2). \tag{4}$$

Here μ is the distortion parameter and $\mathbf{p}_u = (x_u, y_u)$ and $\mathbf{p}_d = (x_d, y_d)$ are the undistorted and distorted positions, respectively. In this paper, the distortion center is fixed to coincide with the principal point of the camera and we set $r_d = \|\mathbf{p}_d\|$. To get a consistent radial distortion independent of image size all images coordinates are initially scaled with a factor of

$$scale = \frac{2}{\max(width, height) - 1}, \tag{5}$$

which maps all image coordinates to be between minus one and one.

3. Pose with Radial Distortion

The problem of solving for radial distortion, focal lengths and pose has eight degrees of freedom; one distortion parameter, focal length, three translation parameters and three rotation angles. To simplify the calculations the inverted focal length is used, and by that the calibration matrix will be,

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/f \end{bmatrix}. \tag{6}$$

This can be done since the camera matrix only is given up to scale. In the following $1/f$ will be substituted by w to simplify the notation.

The rotation is parameterized with quaternions. This gives the following rotation matrix,

$$R=\begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2ab+2cd & a^2-b^2-c^2+d^2 \end{bmatrix}. \tag{7}$$

Last is the translation given by a vector $\mathbf{t} = \begin{bmatrix} x & y & z \end{bmatrix}^T$. Composing these, the camera matrix P will be given according to equation (2). To include the radial distortion in this model the projection will be modeled by,

$$\lambda \begin{bmatrix} x_1 \\ x_2 \\ 1+\mu(x_1^2+x_2^2) \end{bmatrix} = P\mathbf{X}. \tag{8}$$

At this stage the number of unknowns is nine. But since the camera matrix is only defined up to scale, the number of unknowns can be reduced by one by setting the quaternion parameter a equal to one. This will result in that the rotation matrix also will include a scale factor and that the scale of the camera matrix will be fixed. By putting $a = 1$ the possibility of a to vanish is also eliminated. This might look like a problem but since a is a real number the probability of it to be zero is zero. We have also verified in the experiments that this does not cause any problems.

The number of unknowns is now down to eight. Every correspondence between an image point and a world point will now give rise to three equations and one additional unknown. Hence four correspondences are necessary to solve the problem. This is a true minimal case were all equations are necessary and in the next two sections it is explained how to solve this problem.

4. Solving the Minimal Setup

To solve the equations generated by (8) the equation are first simplified by using the freedom in choice of coordinate system. In the three dimensional space any similarity transform can be applied. This freedom is used to put the first point in origin and the second in $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$. In the image, only rotation and scaling is allowed since the focal length is unknown. Due to this, the first point is moved to $\begin{bmatrix} 1 & 0 \end{bmatrix}$. To summarize the following point positions will hold for every problem setup,

$$\mathbf{X}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \ \mathbf{X}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \ \mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}. \tag{9}$$

This choice of coordinate system leads to several simplifications of the problem. First we can express the translation coordinate x in measured image points and the quaternion parameters as follows,

$$x = g_1(a,b,c,d) = \frac{u_1}{u_2}(2ad+2bc) - (a^2+b^2-c^2-d^2). \tag{10}$$

u_1 and u_2 are here the x and y coordinates of the second image point. The second simplification is that y can be set to zero. The last simplification from the choice of the coordinate system is that the product between the inverted focal length and z can be expressed in the quaternion parameters and the distortion parameter according to,

$$zw = g_2(a,b,c,d,\lambda) = x(1+\mu), \tag{11}$$

were x is from equation (10).

The next step is to include the last two point correspondences and the last information from the second point $\mathbf{x}_2$. This is done by eliminating λ in equation (8). The elimination is done by multiplying $P\mathbf{X}$ with the following matrix from the left,

$$B = \begin{bmatrix} 0 & -x_3 & x_2 \\ -x_3 & 0 & x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \tag{12}$$

where $x_3 = 1+\mu(x_1^2+x_2^2)$. This is a rank 2 matrix so not all rows need to be used from the equation $BP\mathbf{X} = 0$. For the second image point only the second row of B is used and for the other two the first and the last row are used. This results in five equations in the five unknowns b, c, d, w and μ. With use of Gröbner basis methods this system of polynomial equations will be solved.

5. Gröbner Basis Solver

To solve the system of polynomial equation Gröbner basis methods are used. Gröbner basis methods have successfully been used to solve several systems of polynomial equations derived from computer vision problem in recent years, *e.g.* [4, 5, 12, 19, 27]. The advantages of using Gröbner basis methods is that if the structure of the system is the same for a large number of problem some calculations can be done symbolically in advance, which leads to an efficient method to solve the systems of polynomial equations.

Only the major concepts of Gröbner basis methods will be described in this paper. For basic background theory we recommend [8] and [7] by Cox *et al.* For details for the use in computer vision see [26] for example.

The first step in constructing a Gröbner basis solver is to find out the number of solutions of the system. This can be done once and will hold for all geometrical setups of the same minimal problem. The method to find the number of solutions is to use a symbolic program *e.g.* Macaulay 2 [13]. The problem of this paper turns out to have 24 solutions with the given formulation. However, it is quadratic in the focal length so it will never give more than 12 geometrically plausible solutions.

The second step is to expand the initial set of equations. This is done by multiplying the initial equations with a set of monomials. This results in more linearly independent equations with the same solution set and by that it is possible to construct the Gröbner basis. In the problem at hand, the two original equations of lowest degree, these resulting from multiplication with the last row of B in equation (12), are multiplied with μ and w. After that all the nine equations, at this stage, are multiplied with all monomials up to degree four in the unknowns. The result of this expansion is 1134 equations and 720 different monomials. This can be written as,

$$C_{\text{exp}} \mathbf{X}_{\text{exp}} = 0, \tag{13}$$

where C_{exp} is a 1134×720 matrix holding all coefficients and $\mathbf{X}_{\text{exp}}$ is a 720 elements long column vector with all occurring monomials. This equation corresponds to equation (4) in [5] and the rest of the solver will follow that paper. Those details will not be given here. Another way to construct the solver is to use the automatic solver generator by Kukelova *et al.* [19]. We chose the first alternative since it enhances the numerics.

The usual step at this stage is to sort the monomials in a monomial order and then find the Gröbner basis by a Gauss-Jordan elimination. Instead the method from [5] is followed to enhance the numerics. This means that QR-factorization with column pivoting is used together with adaptive truncation of the ideal. The truncation threshold used is 10^{-8}.

To construct the action matrix describing multiplication following [5], the permissible monomials and the action variable needs to be given. In this paper we choose all monomials up to degree three to be in the permissible set and b to be the action variable. The number of permissible monomials with the given choice is 56.

With this the action matrix can be constructed and the eigenvectors of the transposed action matrix will hold all solutions to the system, see [5] for details on how to construct the action matrix with the method chosen in this paper.

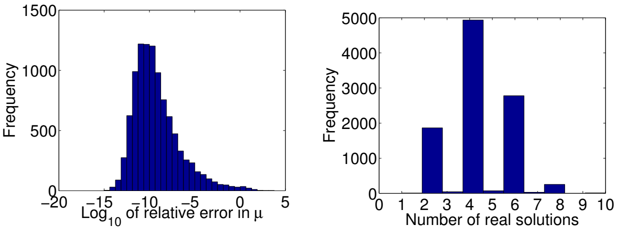

Figure 2. Left: Histogram of errors over 10000 runs on noise free data. Right: Histogram of the number of solutions with real positive focal length found on the same data.

Matlab code for the solver used in this paper is available online at `http://www.maths.lth.se/vision/downloads`.

6. Experiments on Synthetical Data

In this section we study some basic properties of our new algorithm on synthetic data. We start off with a straightforward test on noise free data to check stability and the distribution of plausible solutions. In this experiment, random scenes were generated by drawing four points uniformly from a cube with side length 1000 centered at the origin. A camera was then placed at a distance of 1000 from the origin pointing approximately at the center. The camera was calibrated except for the focal length that was set to around 1000. Radial distortion was then added to the projected points and the distortion parameter was uniformly drawn from the interval $[-0.5, 0]$. Our new minimal solver was run on 10000 such instances and Figure 2 displays the results of this experiment. The numerical error stays very low for almost all cases. A very small number of examples show larger errors, but these do not pose any serious problem since the intended application is RANSAC where lots of instances are solved and only the best one is kept. As previously mentioned, the largest possible number of plausible solutions (real positive focal length) is 12. However, the largest observed number of plausible solutions for the 10000 random instances was 10 and in all but a few exceptions we got 6 solutions or fewer.

To verify that the solver does give accurate results and not just adapts to noise we made an experiment where we measured the relative error in focal length as a function of noise. The setup was the same as in the previous experiment and the standard deviation of the noise was varied between (the equivalent of) zero and three pixels on a 1000×1000 pixel image. For each noise level, 1000 problem instances were tested. The results are given in Table 1 and show that our method is robust to noise. Even with as large errors as three pixels, the median error in focal length is less than seven percent.

The time consumed of the solver was also measured. On an Intel Core 2 with clock rate of 2.13 GHz the average

Noise	Median	75th percentile
0.0	$1.5 \cdot 10^{-11}$	$5.1 \cdot 10^{-10}$
0.5	$1.4 \cdot 10^{-2}$	$4.1 \cdot 10^{-2}$
1.0	$2.3 \cdot 10^{-2}$	$6.8 \cdot 10^{-2}$
2.0	$5.2 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$
3.0	$6.7 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$

Table 1. The relative error of the focal length for different levels of noise. The noise is given in pixels.

time for a call over 1000 tests was 60 ms when a Matlab implementation was used.

The next synthetic experiment was designed to investigate how important it is to include radial distortion in the minimal solver. To do that, a setup with 80 inliers and 120 outliers was constructed. Radial distortion was then added to all image points. Three different levels of radial distortion were used, 0, -0.07 and -0.2. Zero distortion was included to test our algorithm compared to a method that assumes no radial distortion. A distortion of -0.07 was used since the normal lens later used in the real experiments roughly has this distortion. This lens is shipped with a consumer level SLR camera. The last value, -0.2, corresponds to the distortion of the fisheye lens later used in the experiments. Noise corresponding to one pixel in a 1000×1000 image was then added to each image point. On this data a RANSAC step was applied and the number of inliers was counted. In the RANSAC loop a point was considered to be an inlier if the reprojection error was less than 0.01 times the mean value of all coordinates of all points given that the origin is in the center of the image. One hundred individual scenes were used for each distortion level. All distortion levels were tested both on the proposed method and on the method of Bujnak *et al*. [4]. The algorithm of Bujnak *et al*. solves for pose and focal length using four points. The results of this experiment are shown in Figure 3 with increasing radial distortion from left to the right. Our method is plotted with a solid blue line and Bujnak's with dashed red. The results show as expected that if radial distortion is zero it is slightly better not to estimate it. The two other plots show that the use of radial distortion gives a large boost in performance. Note especially the large difference even with a small distortion of a standard SLR camera lens. These results will also be confirmed in the experiments with real data.

7. Experiments on Real Data

The real world experiments were done in a leave one out manner. This was done by first creating a model of a scene using the Photo tourism bundler [25]. To build the model 93 images from a shopping street were used covering around one hundred meters. Example of one of those images is shown to the right in Figure 4. In all these images a regular lens was used. For 29 of these images a second image was taken from the exact same position (a tripod was used to fixate the position) with a fisheye lens. See Figure 4 (left) for an example. Then one image at a time (of those images with a corresponding fisheye image) were removed from the model. The pose of the removed image was estimated using the proposed method both for the fisheye image and the regular image. The positions were then compared with the positions estimated by Photo tourism. Note that Photo tourism does not give an exact solution and the authors do not know the precision, but it will still be used as ground truth in this paper. The results of this experiment was also compared with the method by Bujnak *et al.*

The pose estimation is done with the following method. First SIFT is applied to the image for which the pose should be estimated. The next step is to find potential correspondences in the image. This is done by nearest neighbor. A point is considered a correspondence if the distance to the closest point times 0.9 is not smaller than the distance to the second closest point. When potential correspondences are found, a RANSAC step is performed to find true correspondences. In the end local optimization is performed.

The first evaluation on the real data experiments was to find the number of inliers given the number of RANSAC iterations. The threshold for a point to be considered an inlier is the same as in the corresponding synthetic experiment. In Figure 5 the result is shown. To the left is the result when the fisheye lens is used and to the right is the result for the regular lens. The graphs show an average over one hundred trials for the images shown in Figure 4. The result is typical for most of the images. It is obvious that the use of radial distortion boosts the performance significantly. In some of the tests the method without distortion almost fails to get more inliers than the minimal set. This shows that the use of radial distortion already in the RANSAC step is an important way to increase the performance of the pose estimation. The result is similar to those in the synthetic experiments.

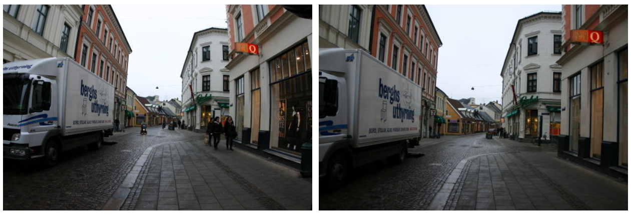

Figure 4. Test images used for the experiment whose results are shown in Figure 5. The images were taken at the exact same position.

The next experiment to evaluate the proposed method is to compare the pose estimated by our new method with the position given by Photo tourism. To do this the inliers, position, focal length and radial distortion given by the

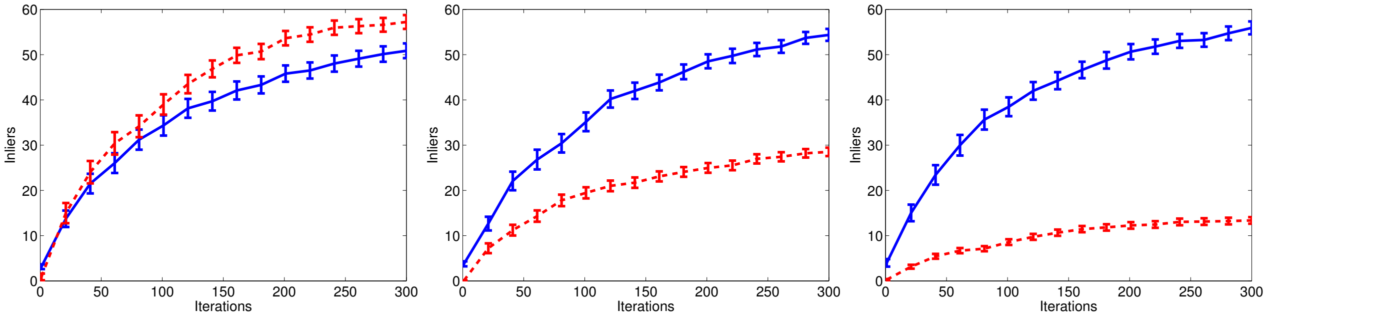

Figure 3. Number of inliers given the number of RANSAC iterations for an example with 80 inliers and 120 outliers. Noise was set to correspond to one pixel in a 1000 × 1000 pixels image. The distortion parameter, μ, was fixed to, from left to right, 0, -0.07, -0.2 and one hundred examples were performed for each level of distortion. The blue solid line is the method of this paper and the dashed red line is the method proposed by Bujnak *et al.*

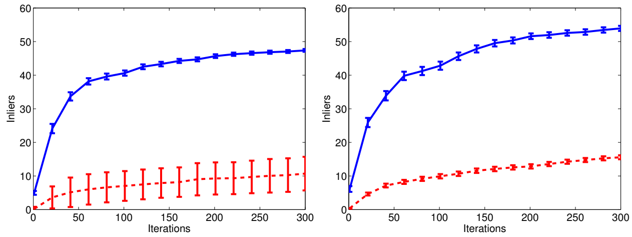

Figure 5. The number of inliers given the number of RANSAC iterations. To the left, a fisheye lens was used and to the right a regular lens was used. The blue solid line is for the method proposed in this paper and the dashed red line is for the method which does not include distortion.

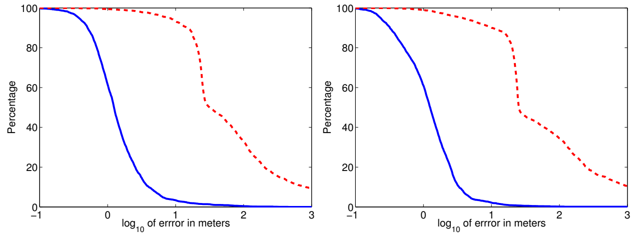

Figure 6. The percentage of images with an estimated position further away then a given distance to the position given by Photo tourism. The error is roughly given in meters. Notice the logarithmic scale. The blue solid line is for the proposed method and the red dashed represents method without distortion. The left plot is for the fisheye lens and the right is for a regular lens.

RANSAC engine are used in a local optimization. The optimization is done for all the unknown parameters. The result is compared with the result when Bujnak's method is used. For that method the same local optimization is performed with the radial distortion initiated with $\mu = 0$. The scale of the model in this experiment is adjusted so that the errors roughly correspond to meters in camera position. Each of the 29 camera positions used in the experiment is estimated one hundred times so the pose estimation has been performed 2900 times. In Figure 6 the result of this experiment is shown. To the left is the result when the fisheye lens is used and to the right is the result for the regular lens.

The precision of Photo tourism that is used for the error measurements is unknown to the authors. Due to that the result for the smallest errors are hard to interpret. We estimate that on this data set, Photo tourism achieves roughly an accuracy of one to a couple of meters. Thus error measurements below that are not reliable. Nevertheless, one can see clearly that our new minimal algorithm gives much more accurate results compared to the previous method which does not take distortion into account in the RANSAC engine.

The results of the pose estimation for the proposed method on a fisheye lens was also compared with the result when the regular lens was used. In Figure 7 the result is shown. In the figure, the blue solid line shows the result with the fisheye lens and the red dashed line shows the result with the regular lens. The plot shows that the amount of radial distortion gives almost no impact on the result.

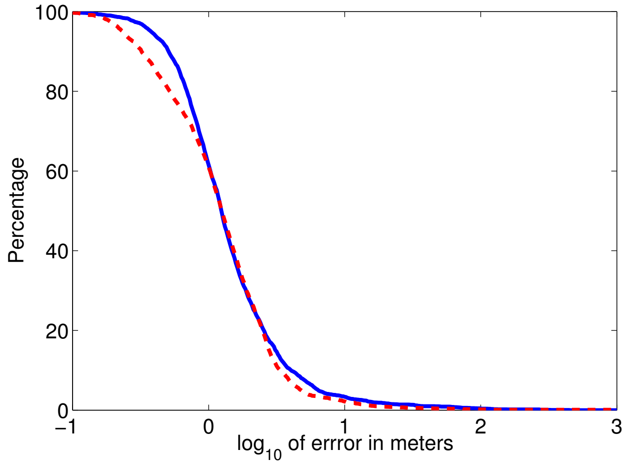

Figure 7. Error in meters for distorted image and non-distorted image using the proposed algorithm on a logarithmic scale. The blue line represents the distorted images and the red show the result for images taken with a regular lens.

The last experiment is a kernel voting experiment where the distorted image in Figure 1 (left) was used. The image was localized 500 times with the minimal solver and the results of the estimations of the radial distortion were used in a kernel voting scheme to find the radial distortion. The results of the kernel voting is shown in Figure 8. The peak of the curve is at $\mu = -0.20$ and that value was used to remove the distortion on the original fisheye image. The undistorted image is shown in Figure 1 (right). Notice how the curved lines in the original image have been straightened in the undistorted image. This shows that the estimated radial distortion is reasonably accurate.

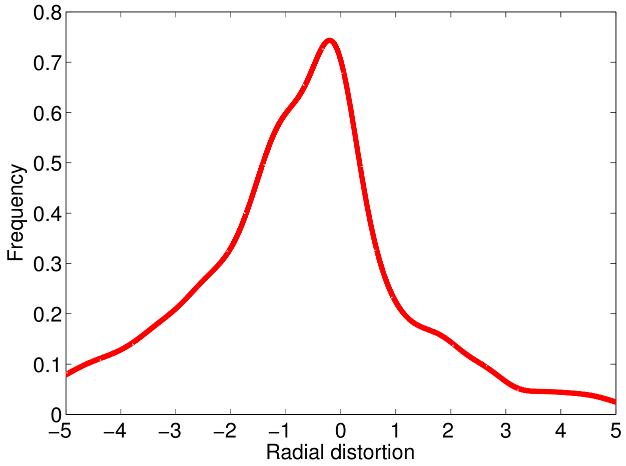

Figure 8. Result of the kernel voting for radial distortion. The standard deviation of the Gaussian kernel was fixed to $1/3$ and the peak of the curve is at $\mu = -0.20$.

8. Conclusions

In this paper a method to estimate the position, rotation, focal length and radial distortion from a minimal set of correspondences to a 3D model is presented. This is the first algorithm presented to do this estimation. The parameterization used in this paper gives a system of polynomial equations. This system is solved with Gröbner basis methods. This gives a fast and numerical stable method that can be used in a RANSAC loop.

Previous methods have not assumed radial distortion in the RANSAC engine and in this paper it is shown that the benefits of using radial distortion in the core of the RANSAC engine is significant. This is shown both on synthetical and real data when a fisheye lens is used. That the improvements are large with the fisheye lens comes as no surprise due to the heavy radial distortion in this case. More surprising are the large improvements for a regular lens of an SLR camera. The reason for this is that there is some radial distortion even in those kind of lenses and evidently, that distortion can have a large effect on the estimated position.

The experiments in the paper also show that the amount of radial distortion has little impact on the result. Hence can the new method be used both when no radial distortion is present and on images with heavy radial distortion.

Acknowledgments

This work has been funded by the European Research Council (GlobalVision grant no. 209480), the Swedish Research Council (grant no. 2007-6476 and 2008-5393) and the Swedish Foundation for Strategic Research (SSF) through the programme Future Research Leaders and the two projects ENGROSS and Wearable Visual Systems.

References

[1] M. Abidi and T. 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, 1995.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *9th European Conference on Computer Vision*, Graz Austria, 2006.

[3] D. C. Brown. Close-range camera calibration. *Photometric Engineering*, 37:855–866, 1971.

[4] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *Proc. Conf. Computer Vision and Pattern Recognition, Anchorage, USA*, 2008.

[5] M. Byröd, K. Josephson, and K. Åström. A column-pivoting based strategy for monomial ordering in numerical gröbner basis calculations. In *The 10th European Conference on Computer Vision*, 2008.

[6] M. Byröd, Z. Kukelova, K. Josephson, T. Pajdla, and K. Åström. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. In *Proc. Conf. Computer Vision and Pattern Recognition, Anchorage, USA*, 2008.

[7] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer Verlag, 1998.

[8] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer, 2007.

[9] F. Devernay and O. D. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. *Investigative and Trial Image Processing*, 2567:62–72, 1995.

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–95, 1981.

[11] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of Computer Vision and Pattern Recognition Conference (CPVR)*, pages 125–132, 2001.

[12] C. Geyer and H. Stewénius. A nine-point algorithm for estimating para-catadioptric fundamental matrices. In *Proc.*

Conf. Computer Vision and Pattern Recognition, Minneapolis, USA, June 2007.

[13] D. Grayson and M. Stillman. Macaulay 2. Available at http://www.math.uiuc.edu/Macaulay2/, 1993-2002. An open source computer algebra software.

[14] J. A. Grunert. Das pothenot'sche problem, in erweiterter gestalt; nebst bemerkungen über seine anwendung in der geodäsie. *Archiv der Mathematik und Physik*, 1:238–248, 1841.

[15] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions for the three point perspective pose estimation problem. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 592–598, 1991.

[16] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. Second Edition.

[17] H. Jin. A three-point minimal solution for panoramic stitching with lens distortion. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[18] K. Josephson, M. Byröd, F. Kahl, and K. Åström. Image-based localization using hybrid feature correspondences. In *The second international ISPRS workshop BenCOS 2007, Towards Benchmarking Automated Calibration, Orientation, and Surface Reconstruction from Images*, 2007.

[19] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *Proc. 10th European Conf. on Computer Vision, Marseille, France*, 2008.

[20] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. In *CVPR*, 2007.

[21] H. Li. A simple solution to the two-view focal-length algorithm. In *Proc. 9th European Conf. on Computer Vision, Graz, Austria*, 2006.

[22] H. Li and R. Hartley. A non-iterative method for lens distortion correction from point matches. In *Workshop on Omnidirectional Vision*, Beijing China, Oct. 2005.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 2004.

[24] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Computing*, 22(10):761–767, 2004.

[25] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 2007.

[26] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, 2005.

[27] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, June 2006.

[28] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, pages 278–284. IEEE Computer Society Press, 1999.