MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1595 November, 1996

Lens Distortion Calibration Using Point Correspondences

Gideon P. Stein
gideon@ai.mit.edu

Abstract

This paper describes a new method for lens distortion calibration using only point correspondences in multiple views, without the need to know either the 3D location of the points or the camera locations. The standard lens distortion model is a model of the deviations of a real camera from the ideal pinhole or projective camera model. Given multiple views of a set of corresponding points taken by ideal pinhole cameras there exist epipolar and trilinear constraints among pairs and triplets of these views. In practice, due to noise in the feature detection and due to lens distortion these constraints do not hold exactly and we get some error. The calibration is a search for the lens distortion parameters that minimize this error. Using simulation and experimental results with real images we explore the properties of this method. We describe the use of this method with the standard lens distortion model, radial and decentering, but it could also be used with any other parametric distortion models. Finally we demonstrate that lens distortion calibration improves the accuracy of 3D reconstruction.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research was provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-94-01-0994. Gideon P. Stein is also supported by Alphatech/ DARPA "Automatic Target Recognition" project 95009-5381 and TASC/ DARPA "MSTAR Match Module" project J-08011-S95042.

1 Introduction:

Radial lens distortion can be a significant factor in medium to wide angle lenses. These are typically the lenses used when performing image based 3D reconstruction of large objects or in a confined space. The errors can be 10-100 pixels at the edges of the image [12]. This paper describes a new method for lens distortion calibration using point correspondences in multiple views without the need to know either the 3D location of the points or the camera locations. In fact one can use the same point correspondence data which is to be used for subsequent vision tasks such as 3D reconstruction. The method is therefore suitable as an online preprocessing stage.

There are two variants of this method. The first uses two images and the second uses three images. I will briefly describe the two methods here. Due to lack of space this paper will focus on the 3 image method.

1.1 The two image method:

Given a point in one view we know that the corresponding point in the other view must line along an epipolar line. Given at least 8 point correspondences we can find the epipoles and epipolar lines in a linear manner (5 using non-linear methods). With more than 8 points we can find the least squares solution.

This constraint holds true for ideal pinhole cameras. Due to noise in the feature detection and due to lens distortion we get some error. The points in the second image do not lie exactly on the epipolar lines. With more than the minimum 8 points we can define a cost function as the RMS of the distances from the points to the corresponding epipolar lines. We then search for the lens distortion parameters that correct the image coordinates of the feature points to minimize this error. The error function is in general well behaved and the distortion parameters can easily be found by nonlinear search techniques.

1.2 The three image method:

Given a set of corresponding points in three images, there exist 4 independent trilinear equations that relate location of the points in the three images [9]. These equations have 27 parameters and given at least 7 point correspondences they can be found in a linear manner. These parameters allow us to reproject corresponding points given in two of the images into the third image.

Again, because of noise and lens distortion, the reprojection is not perfect. We define our cost function to be the RMS reprojection error and search for lens distortion parameters that minimize it.

1.3 What comes next

Section (2) reviews related work. Section (3) reviews the projective constraints on point correspondences in two and three views and describes the radial distortion model. Section (4) spells out the step by step procedure. In Section (5), using simulation, I explore properties of this method and for which camera configurations it works best. Section (6) presents experiments with real images. The method is shown to be robust and to give a significant improvement in precise measurements. Initially I confine the tests to the projective domain and test properties such as planarity and collinearity of points and cross ratios. Finally I perform Euclidean reconstruction of the points. I conclude in section (7) with a discussion of the advantages and drawbacks of this method.

2 Related work

Known world coordinates: The classic method for lens distortion calibration is the bundle adjustment method [8]. It uses one or more views of a calibration object with known 3D coordinates (control points). Using iterative methods it then finds both external and internal camera parameters. The external camera parameters are the position and orientation of each camera. The internal camera parameters include the parameters of the pinhole camera model (principal point, principal distance) and the parameters of lens distortion. Weng et al. [13] also use a known calibration object and iteratively solve for the external and internal parameters including the distortion parameters.

Projective constraints: Under perspective projection, straight lines in space project to straight lines in the image. With real lenses the lines appear instead to be slightly to moderately curved. By searching for lens distortion parameters which straighten the lines the *Plumb Line* method and its derivatives [1] [5] [12] [2] find the lens distortion without needing to find the external parameters or the other internal camera parameters. One or more images can be used.

Unknown world coordinates: Stein [12], and Du and Brady [3] use corresponding points or edges in images where the camera has undergone pure rotation to find the internal camera parameters including lens distortion. The 3D location of the points is not required.

3 Mathematical background

This paper uses results from projective geometry. A very readable introduction to the subject of Projective Geometry is given in the book by Young [14]. A modern book dealing more specifically with the application of Projective Geometry to computer vision is [4].

I will use the following notation. The point in 3D projective space (P^3) will be represented by M_i where the subscript i denotes the i'th point. The subscript i might often be dropped for clarity. I will typically use homogeneous coordinates: $M = (X, Y, Z, T)$, which can be converted to the non-homogeneous coordinates $(\frac{X}{T}, \frac{Y}{T}, \frac{Z}{T})$. The point M_i projects onto the image plane of the j'th camera at point m_{ij} where $m = (x, y, 1)$ or in non-homogeneous image coordinates $m = (x, y)$.

The 4×3 camera projection matrix for the j'th camera will be denoted P_j. Using homogeneous coordinates the perspective projection can be written as:

$$\rho_{ij} m_{ij} = P_j M_i \tag{1}$$

for points $i = 1...n$ in images $j = 1...m$. ρ_{ij} is an unknown scale factor which is different for each point and each image. One can rewrite equation (1) in the following way hiding the scale factor:

$$x_{ij} = \frac{p^j_{11}X_i + p^j_{12}Y_i + p^j_{13}Z_i + p^j_{14}T_i}{p^j_{31}X_i + p^j_{32}Y_i + p^j_{33}Z_i + p^j_{34}T_i} \quad (2)$$

$$y_{ij} = \frac{p^j_{21}X_i + p^j_{22}Y_i + p^j_{23}Z_i + p^j_{24}T_i}{p^j_{31}X_i + p^j_{32}Y_i + p^j_{33}Z_i + p^j_{34}T_i} \quad (3)$$

3.1 The Epipolar Constraint:

We will denote the Fundamental Matrix (Essential Matrix) by F. For every pair of cameras (j,k) there exists a 3×3 Fundamental Matrix F_{jk} such that:

$$m^T_{i,j} F_{jk} m_{i,k} = 0 \quad (4)$$

The above equation is the epipolar constraint. Given a particular camera geometry, embodied in the matrix F, each point in one image defines a line on which the corresponding point in the second image must lie. (The equation of that line is given by $V^T F_{jk} m_{i,k} = 0$.) Given 8 or more point correspondences the Fundamental Matrix F can be determined up to a scale factor using the *eight point algorithm* which is described in [6] with many important implementation details.

3.2 The Trilinear Tensor Constraint:

Shashua [9] shows that given a set of 3D points there exists a set of trilinear equations between the projections of those points into any three perspective views. In total there exist 9 such equations for each point with at most 4 being independent.

Four of the nine equations are as follows:

$$\begin{aligned} x'' \alpha^T_{13} m - x'' x' \alpha^T_{33} m + x' \alpha^T_{31} m - \alpha^T_{11} m = 0 \\ y'' \alpha^T_{13} m - y'' x' \alpha^T_{33} m + x' \alpha^T_{32} m - \alpha^T_{12} m = 0 \\ x'' \alpha^T_{23} m - x'' y' \alpha^T_{33} m + y' \alpha^T_{31} m - \alpha^T_{21} m = 0 \\ y'' \alpha^T_{23} m - y'' y' \alpha^T_{33} m + y' \alpha^T_{32} m - \alpha^T_{22} m = 0 \end{aligned} \quad (5)$$

where x', y' and x'', y'' are the image coordinates in the first and second images respectively. m is the image point in the third view. α_{ij} are column vectors of coefficients. There are a total of nine column vectors α_{ij} for a total of 27 coefficients. Seven point correspondences give 28 equations which are enough to recover the coefficients up to a scale factor. The 27 coefficients can be arranged into a $3 \times 3 \times 3$ tensor [10].

Given 7 points in 3 images one can recover the tensor. Given the tensor and the location of a further corresponding point in two of the images one can calculate the location of that point in the third image.

3.3 The perspective projection model with lens distortion

These projective constraints assume a perfect pinhole model. This is a good model for long focal lengths but medium to wide angle lenses have noticeable lens distortion. The standard model for lens distortion [8] is a mapping from the distorted image coordinates, (x_d, y_d), that are observable, to the undistorted image plane coordinates, (x_u, y_u), which are not physically measurable using the equation:

$$\begin{aligned} x_u &= x_d + x'_d(K_1 r'^2_d + K_2 r'^4_d + \cdots) \\ y_u &= y_d + y'_d(K_1 r'^2_d + K_2 r'^4_d + \cdots) \end{aligned} \quad (6)$$

where K_1 and K_2 are the first and second parameters of radial distortion and:

$$r'^2_d = x'^2_d + y'^2_d = (x^2_d - c_{xr})^2 + (y^2_d - c_{yr})^2 \quad (7)$$

It has been shown in [12] that allowing the center of radial distortion, (c_{xr}, c_{yr}) to be different from the principal point is a good approximation to adding the terms for decentering distortion as given in [8].

4 The algorithm

The step by step algorithm is as follows:

1. Find point correspondences between 3 views.
2. Make an initial guess of the distortion parameters: an appropriate guess for (Cx,Cy) is the center of the image. Choose K1 such that $K1 \times r^3 \leq 0.1$ where r is the distance to the corner of the image.
3. Using the distortion parameters compute the undistorted feature locations (eq.6).
4. Compute the trilinear tensor and then use the tensor to compute the reprojection error for each point (eq.5).
5. Adjust the distortion parameters to reduce the reprojection error. This is done by the LMDIF1 routine (see section 6.1.3).
6. Loop back to step 3 till convergence.

Comments: It is best to assume initially that there is only one parameter of radial distortion that matters (K1). After finding the best K1, one can use that value as a starting guess and try searching for other parameters as well. In cases of very strong distortion or if one has a large number of feature points there might be an advantage to using more parameters.

If there might be different cameras involved one can search for different values of K1 for each camera. If one gets a much smaller RMS reprojection error than when enforcing the same value of K1 for all the cameras then it is a good bet that the cameras are in fact different.

5 Simulation

Simulations were performed to test the effect of camera configurations and noise on the shape of the cost function. Simulations were performed using Matlab and Matlab's random number generator.

Eighty 3D points were generated, uniformly distributed between -1 *units* and 1 *units* in the X,Y and Z directions. These points were projected using a projective camera model with optical axis aligned with the Z direction. The camera was located typically $Z_0 = -4$ *units* from the center of the points with a focal length of $f = 1000\,pixels$. This simulates a camera with a viewing angle of about 55^o.

The 3D points were rotated prior to projecting which is equivalent to the camera rotating around the center of the points at a given radius Z_0. Normally distributed

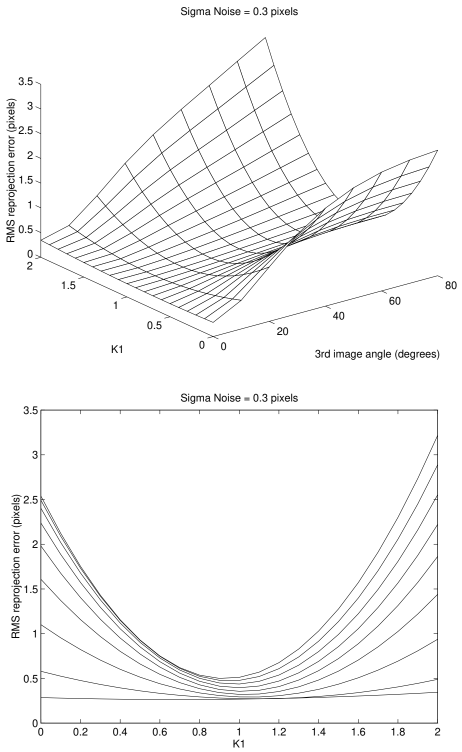


Figure 1: *Simulation results. Reprojection error while varying the radial distortion correction from 0 to 2 times the correct value and varying the viewing angle of the 3rd camera. Gaussian noise of sigma=0.3 was added to feature coordinates.*

random noise was added to the image coordinates which were then distorted using (eq.6) with only one radial distortion parameter, $K_1 = 2.5 \times 10^{-7}$, the value found in section (6).

Consider the following 3 camera setup. Camera 1 is rotated θ^o to the left (around the Y axis). Camera 2 is rotated θ^o to the right. Camera 3 is rotated up (around the X axis) with varying angles $\alpha = 0, 10^o, 20^o \ldots 80^o$. The radial distortion parameter is varied from 0 to $2 \times K_1$ in $0.1 \times K_1$ increments and the reprojection error from camera images 1 and 2 to camera image 3 is calculated.

Due to lack of space I show only the simulation results for $\theta = 20^o$ and additive noise $\sigma = 0.3\ pixel$ (figure 1). Note that for small α the error as a function of K1 is very shallow and it would be hard to find the correct K1 value. With larger noise values the minimum is sometimes outside the plotted range for $\alpha = 0$. Other simulation experiments show that increasing θ towards 40^o the curves become steeper and reducing θ makes them shallower. At $\theta = 2^o$ the results are unreliable even for large α values.

6 Experiments with real images

6.1 Experimental Details

6.1.1 The Images, Features and Feature Detector.

Figures 2a,2b,2c show a typical image triplet used in the experiment. In this case the same camera was used for all three images. The camera motion included a large, unknown, degree of rotation and translation. Figure (3) was taken from approximately the same angle as figure (2c) but the camera was twice as far away and the focal length longer (zoom).

Three lines of feature points were marked on a flat sheet of metal. The features were two small triangles touching at a point. These points are saddle points in the gray level image and result in local minima in the determinant of the Hessian of the image. The point is located to subpixel accuracy by locally fitting the determinant of the Hessian image to a paraboloid and finding the minimum analytically. The location of the points are relatively unaffected by the size of the smoothing filters used in the image processing stage.

The top line, running through the center of the image and the bottom line had 22 points each at 2cm spacing. The middle line had 11 points at 4cm spacing plus an additional point (4th from the left) which was marked not on the metal but on a piece of paper placed on the metal. This special point is therefore not exactly on the same plane as the rest of the points. In order to create a 3D shape we added a few metal objects with clearly defined feature points.

Correspondence was performed manually. In total 75 feature points were detected: 7 points on the top surface of the metal cube, a total of 10 points on the other metal objects and 55 coplanar points in 3 lines plus one nearly coplanar point.

6.1.2 Hardware

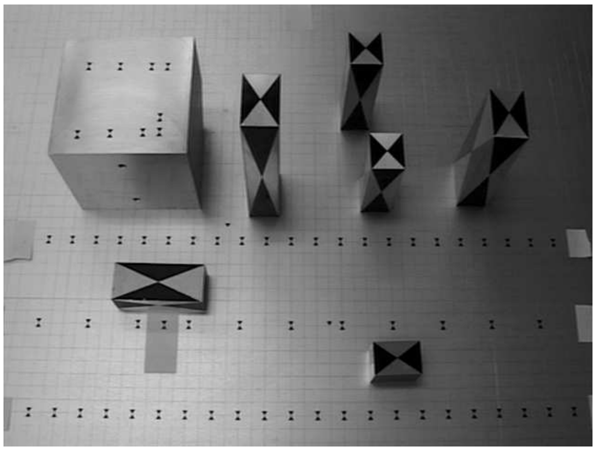
- The camera was a SONY DCR-VX1000, a high quality digital handycam.
- The lens was the built-in zoom lens open to the widest angle giving a corner to corner viewing angle of approximately 55 degrees. This is more or less equivalent to a 35mm lens on a standard 35mm camera or an 8mm lens on 1/3" CCD video camera.
- Image capture and processing was performed on an SGI Indy workstation.

6.1.3 Nonlinear optimization code

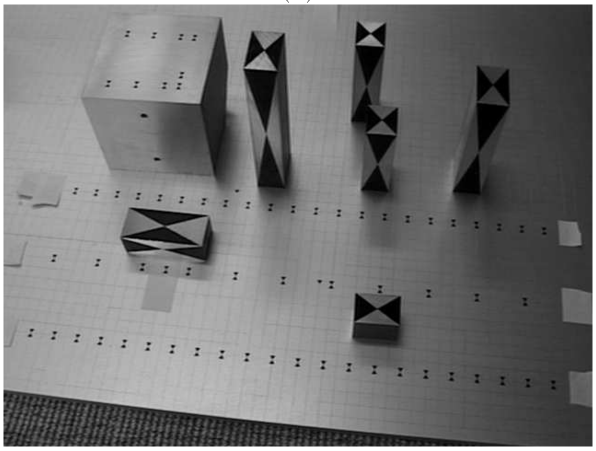
The camera parameters were found using a nonlinear optimization program based on the subroutine LMDIF from the software package MINPACK-1 [7]. This subroutine uses a modified Levenberg-Marquart algorithm.

6.2 Experiment 1: Finding the lens distortion parameters.

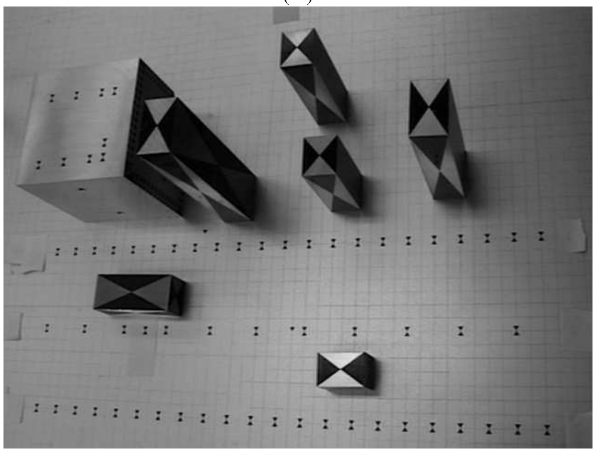
Using all 75 points in the 3 images (figures 2a,2b,2c) I computed the trilinear tensor (see section 3) and then the reprojection error from images 2b and 2c to image 2a. No robust estimation techniques were used. I will denote the radial distortion parameters found as $K1_{best}$,$K2_{best}$,Cx_{best} and Cy_{best}.



(a)



(b)



(c)

Figure 2: *The images used for calibration.*

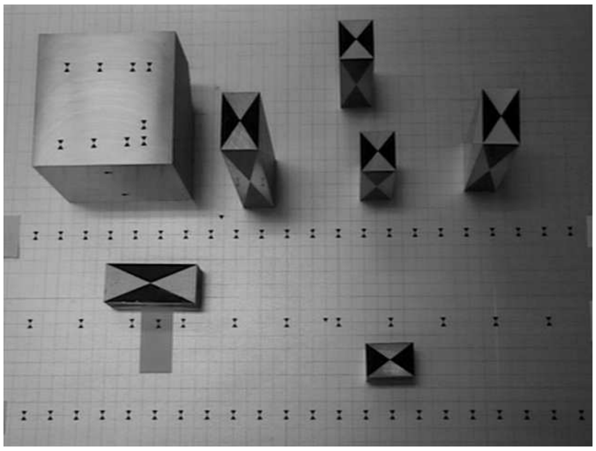


Figure 3: *Alternative third image was taken from further away with a longer focal length. Note that in this case the lines are not curved.*

6.2.1 Convergence range:

One must supply the non-linear search routine with an initial guess. Many initial guesses were tried for the radial distortion parameter (K1). The second parameter (K2) was assumed to be zero and the center of distortion at the image center (320,240).

The non-linear search would converge to a value $K1_{best} = 2.58 \times 10^{-7}$ with a starting guess anywhere in the range $(2.5 \times 10^{-2} : 2.5 \times 10^{-12})$. In other words, through 4 orders of magnitude in either direction. Convergence took between 3 and 10 iterations.

Assuming two radial distortion parameters (K1, K2), the non-linear search would converge to a value $K1_{best} = 4.755 \times 10^{-7} \pm 0.1\%$ and $K2_{best} = -1.153 \times 10^{-12} \pm 0.5\%$ with a starting guess anywhere in the range $K1 = (2.5 \times 10^{-5} : 2.5 \times 10^{-9})$ and $K2 = (-1 \times 10^{-7} : 1 \times 10^{-7})$. In other words, through at least 2 orders of magnitude in either direction of K1. Convergence took between 3 and 10 iterations.

6.2.2 The shape of the cost function:

One of the dangers of using non-linear methods is the presence of local minima. We evaluated the cost function over a range of values of K1 from 0 to $2 \times K1_{best}$ (twice the final estimated value) in $0.1 \times K1_{best}$ increments.

Figure (4) shows the resulting error function. The function is smooth, qualitatively parabolic in shape and with no local minima. These characteristics have repeatedly been seen with no exception in many similar experiments with varying ranges and resolutions.

6.2.3 Assuming different parameters for each camera:

The previous experiments assumed all the images were taken by the same camera but the calibration method can work for different cameras. We can search for 3 separate sets of calibration parameters, one set for each camera.

Allowing for one radial distortion parameter for each camera we get:

$$K1_{best} = (2.55 \times 10^{-7}, 2.69 \times 10^{-7}, 2.19 \times 10^{-7},) \quad (8)$$

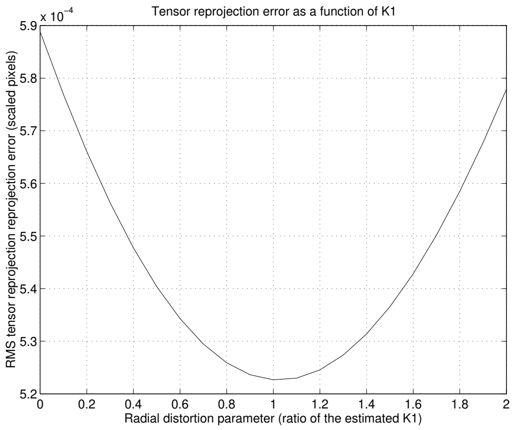


Figure 4: *The tensor reprojection error as a function of the value of the radial distortion K1.*

The 3 images were taken by the same camera so the 3 values for $K1_{best}$ should come out the same. In fact they are within 20% of each other. Better results might be achieved with a larger number of points. Section 7 discusses this issue. Figure (5a) shows the radial distortion correction as function of the radial distance using these three different K1 values.

The third image (Fig: 2c) was then replaced with an image taken from further away using a longer focal length (Fig: 2d). In other words it is a 'different camera'. Allowing for one parameter for radial distortion for each camera the results were

$$K1_{best} = (2.15 \times 10^{-7}, 2.20 \times 10^{-7}, -5.93 \times 10^{-8}) \quad (9)$$

The third camera has much less distortion as we would expect from the longer focal length. Figure (5b) compares the radial distortion correction using the K1 values obtained from the first two images with the correction using the K1 value obtained from the 3rd image. The third camera differs significantly.

6.2.4 Which image is used for reprojection:

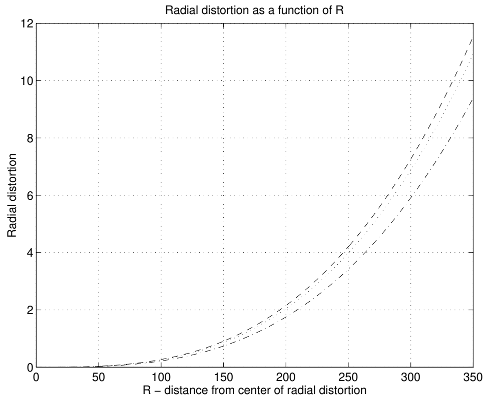
The method has an asymmetry in that images 1 and 2 are used to reproject into image 0. Thus, image 0 is treated differently. All the possible permutations of the 3 images were tried. The results are summarized in table (1). The standard deviation in the value of $K1_{best}$ was 14% which is not large. Nevertheless, to avoid the ambiguity one can use as a cost function the sum of reprojection errors in the 3 distinct possibilities.

6.3 Experiment 2: testing the radial distortion parameters in the projective domain.

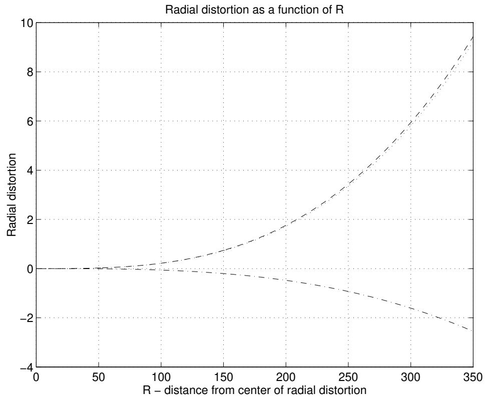
We have no ground truth about the radial distortion parameters. In order to test the radial distortion parameters I used a special set of feature points. 44 of the points are coplanar. They are arranged in 2 straight lines of 22 equally spaced points each.

6.3.1 Collinearity of points

Using one of the images (2a), for each of the two sets of 22 points the image coordinates were fit with a best



(a)



(b)

Legend:	
- - -	Using K1 from 1st camera.
.....	Using K1 from 2nd camera.
_._._	Using K1 from 3rd camera.

Figure 5: *The correction for radial distortion as function of distance from the center of the image. (a) uses 3 K1 values which were obtained from 3 images taken by the same camera. The difference in the correction value is not large. In (b) the 3rd K1 value was obtained using a camera with a longer focal length. The distortion correction is much smaller in that case.*

Table 1: Radial distortion parameter K1 obtained using various ordering of the images.

Order	K1	RMS error
a,b,c	2.58162e-07	e=6.773012e-02
a,c,b	2.5811e-07	e=6.773012e-02
c,a,b	2.4687e-07	e=7.554849e-02
c,b,a	2.46804e-07	e=7.554849e-02
b,a,c	3.19903e-07	e=6.590745e-02
b,c,a	3.19834e-07	e=6.590745e-02
mean	2.7498e-07	
std	3.9313e-08	
std/mean	0.14	

straight line. I then computed the RMS distance of the points from the corresponding lines. This is the measure used in the *plumb line method* described in section (2).

Figure (6a) shows the average distance as a function of the radial distortion parameter K1, where K1 varies from 0.0 to $2.0 \times K1_{best}$ in $0.1 \times K1_{best}$ steps. The function's minimum is at $K1 = K1_{best}$. The average distance of points from the line drops from 1.15 pixels to 0.05 pixels after correction.

6.3.2 Cross ratio of points

The cross ratio of four points along a line is a projective invariant. Taking the 1st, 2nd and 22nd points along each of the lines, one can compute the cross ratio using any of the other points:

$$r_i = \frac{\frac{\alpha_1 - \alpha_2}{\alpha_1 - \alpha_i}}{\frac{\alpha_{22} - \alpha_2}{\alpha_{22} - \alpha_i}} \tag{10}$$

One can invert equation (10) to obtain:

$$\alpha_i = \frac{r_i \alpha_1 (\alpha_{22} - \alpha_2) - r_i \alpha_{22} (\alpha_1 - \alpha_2)}{r_i (\alpha_{22} - \alpha_2) - r_i (\alpha_1 - \alpha_2)} \tag{11}$$

Given that we know that the points along each line are equally spaced we can use equation (11) to estimate the location of points 3 through 21 given the location of points 1, 2 and 22. The difference between the estimated location and the actual measured location of the points is due to feature detection error and also radial distortion. Figure (6b) shows the RMS error in the estimation of all the 38 points as a function of K1. The minimum error is obtained at a value of K1 close ($0.8 \times K1_{best}$) to that obtained using the new calibration method.

6.3.3 Planar mapping of points between two images

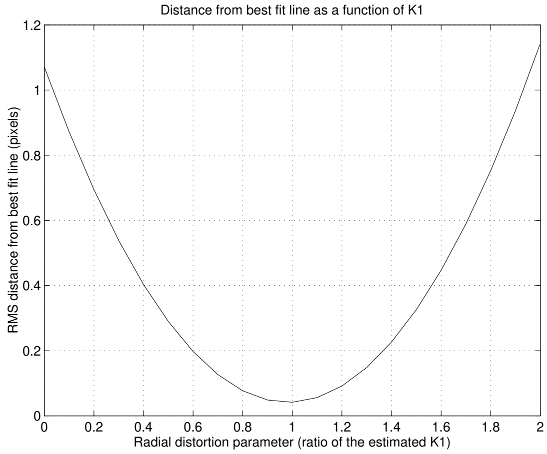
Given two sets of image points obtained from two views of a planar object one can compute a transformation from one set of points to the other which takes the form:

$$m' = Am \tag{12}$$

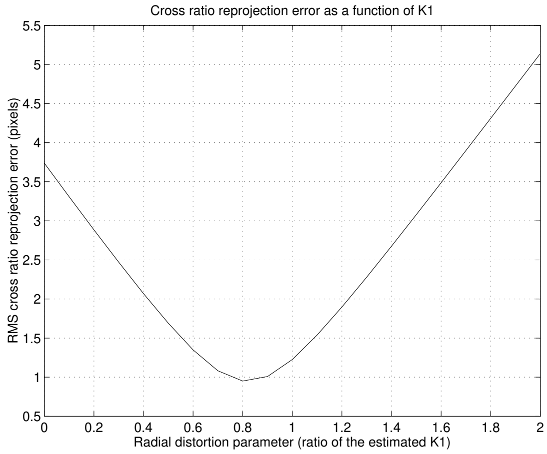
Where m and m' are the corresponding image points in the first and second images and A is a 3×3 transformation matrix. Given more than 4 points one can compute A in a least squares manner. One can then use the matrix A to map all the planar points from one image to the second. Because of noise and radial distortion, the planar mapping reprojection is not perfect. Figure (6c) shows the RMS reprojection error using the images Fig. 2a and Fig. 2b. The minimum error is obtained near the value of K1 found using the distortion calibration.

6.4 Experiment 3: Euclidean reconstruction.

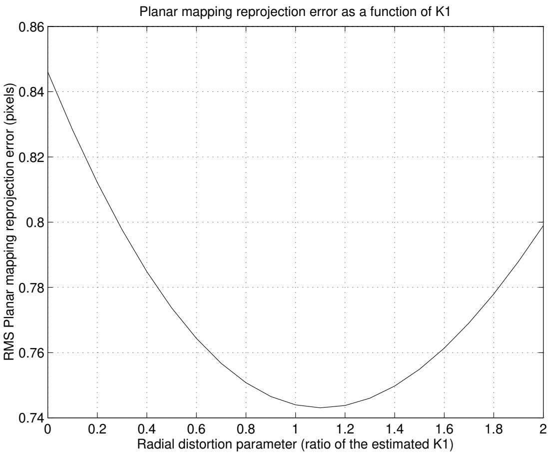
Projective reconstruction from the three images was performed according to [11]. Transformation to Euclidean 3D coordinates was required 5 control points. Three points were chosen from the planar surface and two other points were chosen such that the five were in *general position*, no 4 points coplanar. The world coordinate system was chosen such that the planar surface was the XY plane and the height above the plane was the Z coordinate.



(a)



(b)



(c)

Figure 6: *The error in other projective measures as a function of the value of the radial distortion K1.*

The reconstruction was performed twice, once with the original feature coordinates and once with the corrected coordinates using $K1 = 2.58 \times 10^{-7}$. The best plane was found using the points which came from the planar surface. Figures (7a, 7b, 7c) show the height of the points from the best plane for the three lines. In figure (7c) one can see that for the line close to the edge of the image the line is distinctly curved in depth and that correcting for radial distortion removes the curvature. The curve along the whole length of the line is about $0.3cm$ which is about 1% of the total length of the line ($42cm$). The curvature of the lines closer to the center of the image is smaller.

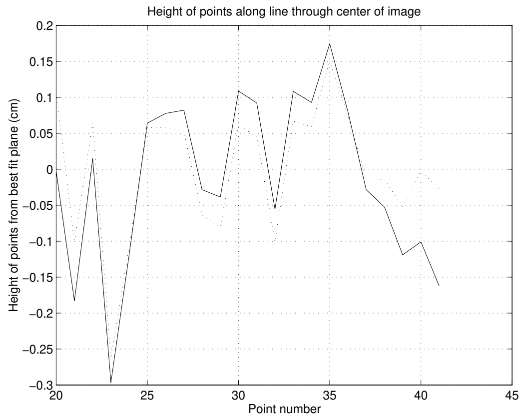
Point number 45 in Figure (7b) is the point that was drawn on the piece of paper and was in fact not coplanar with the other points. In the case of the corrected points reconstruction (dotted line) it is clearly stands apart.

7 Discussion

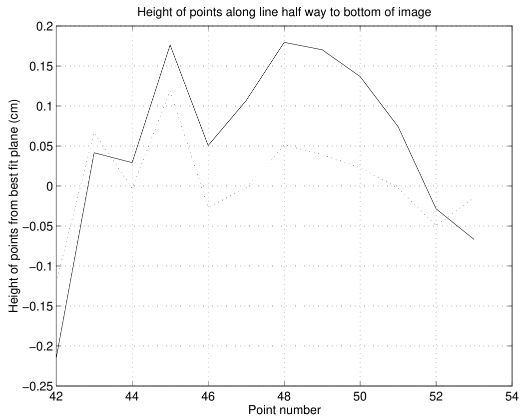
A new method for determining the lens distortion has been described. Although the method is iterative it has been shown that the method converges fast and that the range of convergence is very wide. The resulting distortion parameters are reasonable and have been shown to agree with those determined by other methods.

Since this method uses only point correspondences in multiple views it fits very naturally inside feature based 3D reconstruction systems. It has been shown here that in the case of a 55^o lens, when the distortion is hardly noticeable in the image, radial distortion can cause a small but noticeable error in reconstruction results. Since the computational overhead of this method is small it makes sense to incorporate this method in all 3D reconstruction systems which use medium and wide angle lenses and where accuracy is desired. Especially, since most of the computational machinery required for tasks such as finding the trilinear tensor or fundamental matrix probably already exists in the overall system.

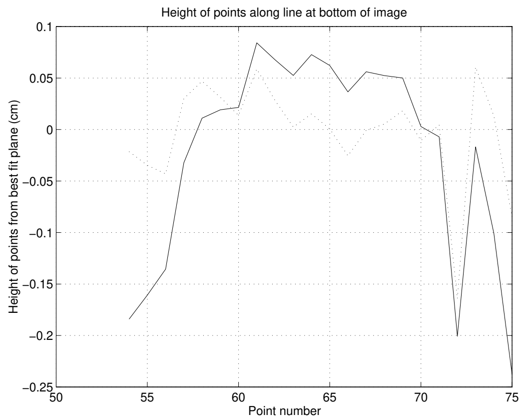
The Manual of Photogrammetry [8] warns us that "the strong coupling that exists between interior elements of orientation [principal point, focal length] and exterior elements can be expected to result in unacceptably large variances for these particular projective parameters when recovered on a frame-by-frame basis". This is also applicable to lens distortion parameters and it was seen in sections (6.2.3) and (6.2.4). Note that the interior and exterior parameters are determined implicitly in the fundamental matrix or the trilinear tensor. To determine the actual lens distortion parameters one can use many images from a variety of locations and apply this method to many triplets of those images. The computational complexity of the method is linear in the number of triplets and number of feature points.



(a)



(b)



(c)

Legend:	
- - -	Reconstruction using uncorrected points.
.....	Reconstruction using corrected points.

Figure 7: *The height of the reconstructed 3D points from the best fit plane. Figures (a), (b) and (c) are the points along the lines which pass close to the center of the image, half way down and close to the bottom, respectively.*

References

[1] **Brown, D.C.**,"Close -range Camera Calibration" *Photogrammetric Engineering* **37** 855-866 (1971)

[2] **Devernay, F. and Faugeras, O.**, "Automatic calibration and removal of distortion from scenes of

structured environments", In *Proceedings of SPIE Conference*, San Diego, CA, July (1995)

[3] **Du, F. and Brady, M.**, "Self Calibration of the Intrinsic Parameters of Cameras for Active Vision Systems" In *Proceedings of CVPR 93*, 477-482, New York, NY, June (1993)

[4] **Faugeras, O.**, *Three Dimensional Computer Vision: a Geometric Viewpoint* , MIT Press, (1993).

[5] **Fryer, J.G. and Mason, S.O.**,"Rapid Lens Calibration of a Video Camera" *Photogrammetric Engineering and Remote Sensing* **55** 437-442, (1989)

[6] **Hartley, R**, "In Defense of the Eight Point Algorithm", In *Proc. of ICCV 95*, 1064-1070, Boston, MA, USA, June (1995)

[7] **More, J.J., et al.**, "User Guide for Minpack-1" Argonne National Laboratory, Argonne, Illinois (1980)

[8] **Slama, C.C. ed**, *Manual of Photogrammetry*,4th edition, American Society of Photogrammetry (1980).

[9] **Shashua, A.** "Algebraic Functions for Recognition", *IEEE Trans. PAMI* **17**, 779-789, (1995)

[10] **Shashua, A. and Werman, M.**, "Trilinearity of Three Perspective Views and its Associated Tensor", In *Proceedings of ICCV 95*, 920-925 Boston, MA, USA, June (1995)

[11] **Shashua, A. and Navab, N.**, "Relative Affine Structure: Canonical Model for 3D from 2D Geometry and Applications." *IEEE Trans. PAMI* (PAMI) Vol. 18(9), pp. 873–883, (1996).

[12] **Stein, G.P.**, "Internal Camera Calibration using Rotation and Geometric Shapes" AITR-1426, Master's Thesis, Massachussets Institute of Technology, Artificial Intelligence Laboratory (1993).

[13] **Weng, J et al.** "Camera Calibration with Distortion Models and Accuracy Evaluation" *IEEE Trans. PAMI* **14**,965-980 (1992)

[14] **Young, J.W.**, *Projective Geometry*, The Mathematical Society of America, (1930)