# Flexible Camera Calibration By Viewing a Plane From Unknown Orientations

Zhengyou Zhang
Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA
zhang@microsoft.com http://research.microsoft.com/~zhang

## Abstract

*We propose a flexible new technique to easily calibrate a camera. It only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. Either the camera or the planar pattern can be freely moved. The motion need not be known. Radial lens distortion is modeled. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques which use expensive equipment such as two or three orthogonal planes, the proposed technique is easy to use and flexible. It advances 3D computer vision one step from laboratory environments to real world use. The corresponding software is available from the author's Web page.*

**Keywords:** Camera Calibration, Intrinsic Parameters, Lens Distortion, Flexible Plane-Based Calibration, Motion Analysis, Model Acquisition.

## 1. Motivations

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. Much work has been done, starting in the photogrammetry community (see [2, 4] to cite a few), and more recently in computer vision ([9, 8, 20, 7, 23, 21, 15, 6, 19] to cite a few). We can classify those techniques roughly into two categories:

**Photogrammetric calibration.** Calibration is performed by observing a calibration object whose geometry in 3-D space is known with very good precision. Calibration can be done very efficiently [5]. The calibration object usually consists of two or three planes orthogonal to each other. Sometimes, a plane undergoing a precisely known translation is also used [20]. These approaches require an expensive calibration apparatus, and an elaborate setup.

**Self-calibration.** Techniques in this category do not use any calibration object. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints [15] on the cameras' internal parameters from one camera displacement by using image information alone. Therefore, if images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity [14, 12]. While this approach is very flexible, it is not yet mature [1]. Because there are many parameters to estimate, we cannot always obtain reliable results.

**Other techniques** exist: vanishing points for orthogonal directions [3, 13], and calibration from pure rotation [10, 18].

Our current research is focused on a desktop vision system (DVS) since the potential for using DVSs is large. Cameras are becoming cheap and ubiquitous. A DVS aims at the general public, who are not experts in computer vision. A typical computer user will perform vision tasks only from time to time, so will not be willing to invest money for expensive equipment. Therefore, flexibility, robustness and low cost are important. The camera calibration technique described in this paper was developed with these considerations in mind.

The proposed technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. The pattern can be printed on a laser printer and attached to a "reasonable" planar surface (e.g., a hard book cover). Either the camera or the planar pattern can be moved by hand. The motion need not be known. The proposed approach lies between the photogrammetric calibration and self-calibration, because we use 2D metric information rather than 3D or purely implicit one. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques, the proposed technique is considerably more flexible. Compared with self-calibration, it gains considerable degree of robustness. We believe the new technique advances 3D computer vision one step from laboratory environments to the real world.

Note that Bill Triggs [19] recently developed a self-calibration technique from at least 5 views of a planar scene. His technique is more flexible than ours, but has difficulty to initialize. Liebowitz and Zisserman [13] described a technique of metric rectification for perspective images of planes using metric information such as a known angle, two equal though unknown angles, and a known length ratio. They also mentioned that calibration of the internal camera parame-

ters is possible provided such metrically rectified planes, although no algorithm or experimental results were shown.

The paper is organized as follows. Section 2 describes the basic constraints from observing a single plane. Section 3 describes the calibration procedure. We start with a closed-form solution, followed by nonlinear optimization. Radial lens distortion is also modeled. Section 4 studies configurations in which the proposed calibration technique fails. It is very easy to avoid such situations in practice. Section 5 provides the experimental results. Both computer simulation and real data are used to validate the proposed technique. In the Appendix, we provides a number of details, including the techniques for estimating the homography between the model plane and its image.

## 2. Basic Equations

We examine the constraints on the camera's intrinsic parameters provided by observing a single plane. We start with the notation used in this paper.

### 2.1. Notation

A 2D point is denoted by $\mathbf{m} = [u, v]^T$. A 3D point is denoted by $\mathtt{M} = [X, Y, Z]^T$. We use $\widetilde{\mathbf{x}}$ to denote the augmented vector by adding 1 as the last element: $\widetilde{\mathbf{m}} = [u, v, 1]^T$ and $\widetilde{\mathtt{M}} = [X, Y, Z, 1]^T$. A camera is modeled by the usual pinhole: the relationship between a 3D point $\mathtt{M}$ and its image projection $\mathbf{m}$ is given by

$$s\widetilde{\mathbf{m}} = \mathbf{A}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\widetilde{\mathtt{M}} \qquad \text{with } \mathbf{A} = \begin{bmatrix}\alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1\end{bmatrix} \tag{1}$$

where $s$ is an arbitrary scale factor; $(\mathbf{R}, \mathbf{t})$, called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to the camera coordinate system; $\mathbf{A}$ is called the camera intrinsic matrix, and $(u_0, v_0)$ are the coordinates of the principal point, $\alpha$ and $\beta$ the scale factors in image $u$ and $v$ axes, and $c$ the parameter describing the skewness of the two image axes.

We use the abbreviation $\mathbf{A}^{-T}$ for $(\mathbf{A}^{-1})^T$ or $(\mathbf{A}^T)^{-1}$.

### 2.2. Homography between the model plane and its image

Without loss of generality, we assume the model plane is on $Z = 0$ of the world coordinate system. Let's denote the $i^{\text{th}}$ column of the rotation matrix $\mathbf{R}$ by $\mathbf{r}_i$. From (1), we have

$$s\begin{bmatrix}u \\ v \\ 1\end{bmatrix} = \mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}\end{bmatrix}\begin{bmatrix}X \\ Y \\ 0 \\ 1\end{bmatrix} = \mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}\begin{bmatrix}X \\ Y \\ 1\end{bmatrix}$$

By abuse of notation, we still use $\mathtt{M}$ to denote a point on the model plane, but $\mathtt{M} = [X, Y]^T$ since $Z$ is always equal to 0. In turn, $\widetilde{\mathtt{M}} = [X, Y, 1]^T$. Therefore, a model point $\mathtt{M}$ and its image $\mathbf{m}$ is related by a homography $\mathbf{H}$:

$$s\widetilde{\mathbf{m}} = \mathbf{H}\widetilde{\mathtt{M}} \qquad \text{with} \quad \mathbf{H} = \mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}. \tag{2}$$

As is clear, the $3 \times 3$ matrix $\mathbf{H}$ is defined up to a scale factor.

### 2.3. Constraints on the intrinsic parameters

Given an image of the model plane, an homography can be estimated (see Appendix A). Let's denote it by $\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}$. From (2), we have

$$\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} = \lambda\mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix},$$

where $\lambda$ is an arbitrary scalar. Using the knowledge that $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal, we have

$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 = 0 \tag{3}$$

$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_1 = \mathbf{h}_2^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 . \tag{4}$$

These are the two basic constraints on the intrinsic parameters, given one homography. Because a homography has 8 degrees of freedom and there are 6 extrinsic parameters (3 for rotation and 3 for translation), we can only obtain 2 constraints on the intrinsic parameters.

## 3. Solving Camera Calibration

This section provides the details how to effectively solve the camera calibration problem. We start with an analytical solution, followed by a nonlinear optimization technique based on the maximum likelihood criterion. Finally, we take into account lens distortion, giving both analytical and nonlinear solutions.

### 3.1. Closed-form solution

Let

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} \equiv \begin{bmatrix}B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33}\end{bmatrix}$$

$$= \begin{bmatrix}\frac{1}{\alpha^2} & -\frac{c}{\alpha^2\beta} & \frac{cv_0 - u_0\beta}{\alpha^2\beta} \\ -\frac{c}{\alpha^2\beta} & \frac{c^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{cv_0 - u_0\beta}{\alpha^2\beta} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(cv_0 - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1\end{bmatrix}. \tag{5}$$

Note that $\mathbf{B}$ is symmetric, defined by a 6D vector

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T . \tag{6}$$

(It actually describes the image of the absolute conic.)

Let the $i^{\text{th}}$ column vector of $\mathbf{H}$ be $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Then, we have

$$\mathbf{h}_i^T\mathbf{B}\mathbf{h}_j = \mathbf{v}_{ij}^T\mathbf{b} \tag{7}$$

$$\text{with} \quad \mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T .$$

Therefore, the two fundamental constraints (3) and (4), from a given homography, can be rewritten as 2 homogeneous equations in $\mathbf{b}$:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} . \tag{8}$$

If $n$ images of the model plane are observed, by stacking $n$ such equations as (8) we have

$$\mathbf{V}\mathbf{b} = \mathbf{0} , \tag{9}$$

where $\mathbf{V}$ is a $2n \times 6$ matrix. If $n \geq 3$, we will have in general a unique solution $\mathbf{b}$ defined up to a scale factor. If $n = 2$, we can impose the skewless constraint $c = 0$, i.e., $[0, 1, 0, 0, 0, 0]\mathbf{b} = 0$, which is added as an additional equation to (9). The solution to (9) is well known as the eigenvector of $\mathbf{V}^T\mathbf{V}$ associated with the smallest eigenvalue (equivalently, the right singular vector of $\mathbf{V}$ associated with the smallest singular value).

Once $\mathbf{b}$ is estimated, we can compute the camera intrinsic matrix $\mathbf{A}$. See Appendix B for the details.

Once $\mathbf{A}$ is known, the extrinsic parameters for each image is readily computed. From (2), we have

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1}\mathbf{h}_1, \ \mathbf{r}_2 = \lambda \mathbf{A}^{-1}\mathbf{h}_2, \ \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2, \ \mathbf{t} = \lambda \mathbf{A}^{-1}\mathbf{h}_3$$

with $\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|$. Of course, because of noise in data, the so-computed matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ does not in general satisfy the properties of a rotation matrix. Appendix C describes a method to estimate the best rotation matrix from a general $3 \times 3$ matrix.

### 3.2. Maximum likelihood estimation

The above solution is obtained through minimizing an algebraic distance which is not physically meaningful. We can refine it through maximum likelihood inference.

We are given $n$ images of a model plane and there are $m$ points on the model plane. Assume that the image points are corrupted by independent and identically distributed noise. The maximum likelihood estimate can be obtained by minimizing the following functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)\|^2 , \tag{10}$$

where $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)$ is the projection of point $\mathtt{M}_j$ in image $i$, according to equation (2). A rotation $\mathbf{R}$ is parameterized by a vector of 3 parameters, denoted by $\mathbf{r}$, which is parallel to the rotation axis and whose magnitude is equal to the rotation angle. $\mathbf{R}$ and $\mathbf{r}$ are related by the Rodrigues formula [5]. Minimizing (10) is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [16]. It requires an initial guess of $\mathbf{A}, \{\mathbf{R}_i, \mathbf{t}_i | i = 1..n\}$ which can be obtained using the technique described in the previous subsection.

### 3.3. Dealing with radial distortion

Up to now, we have not considered lens distortion of a camera. However, a desktop camera usually exhibits significant lens distortion, especially radial distortion. In this section, we only consider the first two terms of radial distortion. The reader is referred to [17, 2, 4, 23] for more elaborated models. Based on the reports in the literature [2, 20, 22], it is likely that the distortion function is totally dominated by the radial components, and especially dominated by the first term. It has also been found that any more elaborated modeling not only would not help (negligible when compared with sensor quantization), but also would cause numerical instability [20, 22].

Let $(u, v)$ be the ideal (nonobservable distortion-free) pixel image coordinates, and $(\breve{u}, \breve{v})$ the corresponding real observed image coordinates. Similarly, $(x, y)$ and $(\breve{x}, \breve{y})$ are the ideal (distortion-free) and real (distorted) normalized image coordinates. We have [2, 22]

$$\begin{aligned} \breve{x} &= x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \breve{y} &= y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] , \end{aligned}$$

where $k_1$ and $k_2$ are the coefficients of the radial distortion. The center of the radial distortion is the same as the principal point. From $\breve{u} = u_0 + \alpha\breve{x} + c\breve{y}$ and $\breve{v} = v_0 + \beta\breve{y}$, we have

$$\breve{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{11}$$

$$\breve{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] . \tag{12}$$

**Estimating Radial Distortion by Alternation.** As the radial distortion is expected to be small, one would expect to estimate the other five intrinsic parameters, using the technique described in Sect. 3.2, reasonable well by simply ignoring distortion. One strategy is then to estimate $k_1$ and $k_2$ after having estimated the other parameters. Then, from (11) and (12), we have two equations for each point in each image:

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \breve{u}-u \\ \breve{v}-v \end{bmatrix} .$$

Given $m$ points in $n$ images, we can stack all equations together to obtain in total $2mn$ equations, or in matrix form as $\mathbf{D}\mathbf{k} = \mathbf{d}$, where $\mathbf{k} = [k_1, k_2]^T$. The linear least-squares solution is given by

$$\mathbf{k} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{d} . \tag{13}$$

Once $k_1$ and $k_2$ are estimated, one can refine the estimate of the other parameters by solving (10) with $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)$ replaced by (11) and (12). We can alternate these two procedures until convergence.

**Complete Maximum Likelihood Estimation.** Experimentally, we found the convergence of the above alternation technique is slow. A natural extension to (10) is then to estimate the complete set of parameters by minimizing the following functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{m}_{ij} - \breve{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)\|^2 \ , \qquad (14)$$

where $\breve{\mathbf{m}}(\mathbf{A}, k1, k2, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)$ is the projection of point $\mathtt{M}_j$ in image $i$ according to equation (2), followed by distortion according to (11) and (12). This is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [16]. A rotation is again parameterized by a 3-vector $\mathbf{r}$, as in Sect. 3.2. An initial guess of $\mathbf{A}$ and $\{\mathbf{R}_i, \mathbf{t}_i | i = 1..n\}$ can be obtained using the technique described in Sect. 3.1 or in Sect. 3.2. An initial guess of $k_1$ and $k_2$ can be obtained with the technique described in the last paragraph, or simply by setting them to 0.

### 3.4. Summary

The recommended calibration procedure is as follows:

1. Print a pattern and attach it to a planar surface;
2. Take a few images of the model plane under different orientations by moving either the plane or the camera;
3. Detect the feature points in the images;
4. Estimate the five intrinsic parameters and all the extrinsic parameters using the closed-form solution as described in Sect. 3.1;
5. Estimate the coefficients of the radial distortion by solving the linear least-squares (13);
6. Refine all parameters by minimizing (14).

## 4. Degenerate Configurations

We study in this section configurations in which additional images do not provide more constraints on the camera intrinsic parameters. Because (3) and (4) are derived from the properties of the rotation matrix, if $\mathbf{R}_2$ is not independent of $\mathbf{R}_1$, then image 2 does not provide additional constraints. In particular, if a plane undergoes a pure translation, then $\mathbf{R}_2 = \mathbf{R}_1$ and image 2 is not helpful for camera calibration. In the following, we consider a more complex configuration.

**Proposition 1.** *If the model plane at the second position is parallel to its first position, then the second homography does not provide additional constraints.*

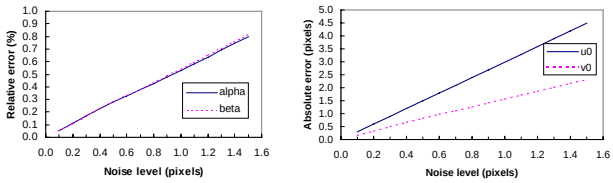


Figure 1: Errors vs. the noise level of the image points

The proof is omitted due to space limitation, and is available from our technical report [24]. In practice, it is very easy to avoid the degenerate configuration: we only need to change the orientation of the model plane from one snapshot to another.

## 5. Experimental Results

The proposed algorithm has been tested on both computer simulated data and real data. The closed-form solution involves finding a singular value decomposition of a small $2n \times 6$ matrix, where $n$ is the number of images. The nonlinear refining within the Levenberg-Marquardt algorithm takes 3 to 5 iterations to converge.

### 5.1. Computer Simulations

The simulated camera has the following property: $\alpha = 1250$, $\beta = 900$, $c = 1.09083$ (equivalent to 89.95°), $u_0 = 255$, $v_0 = 255$. The image resolution is $512 \times 512$. The model plane is a checker pattern containing $10 \times 14 = 140$ corner points (so we usually have more data in the $v$ direction than in the $u$ direction). The size of pattern is 18cm×25cm. The orientation of the plane is represented by a 3D vector $\mathbf{r}$, which is parallel to the rotation axis and whose magnitude is equal to the rotation angle. Its position is represented by a 3D vector $\mathbf{t}$ (unit in centimeters).

**Performance w.r.t. the noise level.** In this experiment, we use three planes with $\mathbf{r}_1 = [20°, 0, 0]^T$, $\mathbf{t}_1 = [-9, -12.5, 500]^T$, $\mathbf{r}_2 = [0, 20°, 0]^T$, $\mathbf{t}_2 = [-9, -12.5, 510]^T$, $\mathbf{r}_3 = \frac{1}{\sqrt{5}}[-30°, -30°, -15°]^T$, $\mathbf{t}_3 = [-10.5, -12.5, 525]^T$. Gaussian noise with 0 mean and $\sigma$ standard deviation is added to the projected image points. The estimated camera parameters are then compared with the ground truth. We measure the relative error for $\alpha$ and $\beta$, and absolute error for $u_0$ and $v_0$. We vary the noise level from 0.1 pixels to 1.5 pixels. For each noise level, we perform 100 independent trials, and the results shown are the average. As we can see from Fig. 1, errors increase linearly with the noise level. (The error for $c$ is not shown, but has the same property.) For $\sigma = 0.5$ (which is larger than the normal noise in practical calibration), the errors in $\alpha$ and $\beta$ are less than 0.3%, and the errors in $u_0$ and $v_0$ are around 1 pixel. The error in $u_0$ is larger than that in $v_0$. The main reason is that there are less data in the $u$ direction than in the $v$ direction, as we said before.

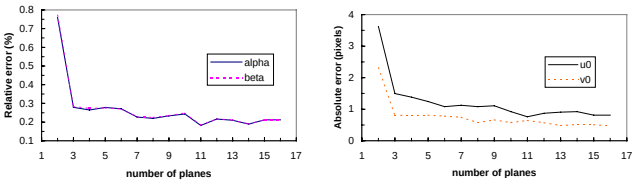


Figure 2: Errors vs. the number of images of the model plane

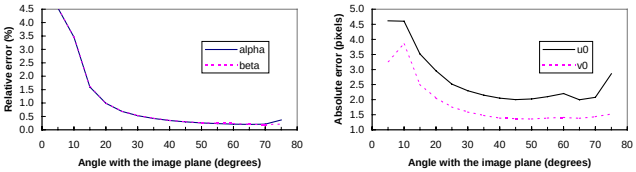


Figure 3: Errors vs. the angle of the model plane w.r.t. the image plane

**Performance w.r.t. the number of planes.** This experiment investigates the performance with respect to the number of planes (more precisely, the number of images of the model plane). The orientation and position of the model plane for the first three images are the same as in the last subsection. From the fourth image, we first randomly choose a rotation axis in a uniform sphere, then apply a rotation angle of 30°. We vary the number of images from 2 to 16. For each number, 100 trials of independent plane orientations (except for the first three) and independent noise with mean 0 and standard deviation 0.5 pixels are conducted. The average result is shown in Fig. 2. The errors decrease when more images are used. From 2 to 3, the errors decrease significantly.

**Performance w.r.t. the orientation of the model plane.** This experiment examines the influence of the orientation of the model plane with respect to the image plane. Three images are used. The orientation of the plane is chosen as follows: the plane is initially parallel to the image plane; a rotation axis is randomly chosen from a uniform sphere; the plane is then rotated around that axis with angle $\theta$. Gaussian noise with mean 0 and standard deviation 0.5 pixels is added to the projected image points. We repeat this process 100 times and compute the average errors. The angle $\theta$ varies from 5° to 75°, and the result is shown in Fig. 3. When $\theta = 5°$, 40% of the trials failed because the planes are almost parallel to each other (degenerate configuration), and the result shown has excluded those trials. Best performance seems to be achieved with an angle around 45°. Note that in practice, when the angle increases, foreshortening makes the corner detection less precise, but this is not considered in this experiment.

### 5.2. Real Data

The proposed technique is now routinely used in our vision group and also in the graphics group. Here, we provide the result with one example.

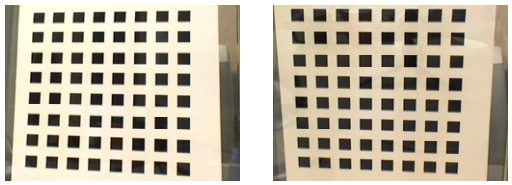

Figure 5: First and second images after having corrected radial distortion

The camera to be calibrated is an off-the-shelf PULNiX CCD camera with 6 mm lens. The image resolution is 640×480. The model plane contains a pattern of $8 \times 8$ squares, so there are 256 corners. The size of the pattern is 17cm×17cm. Five images of the plane under different orientations were taken, as shown in Fig. 4. We can observe a significant lens distortion in the images. The corners were detected as the intersection of straight lines fitted to each square.

We applied our calibration algorithm to the first 2, 3, 4 and all 5 images. The results are shown in Table 1. For each configuration, three columns are given. The first column (`initial`) is the estimation of the closed-form solution. The second column (`final`) is the maximum likelihood estimation (MLE), and the third column ($\sigma$) is the estimated standard deviation, representing the uncertainty of the final result. As is clear, the closed-form solution is reasonable, and the final estimates are very consistent with each other whether we use 2, 3, 4 or 5 images. We also note that the uncertainty of the final estimate decreases with the number of images. The last row of Table 1, indicated by `RMS`, displays the root of mean squared distances, in pixels, between detected image points and projected ones. The MLE improves considerably this measure.

The careful reader may remark the inconsistency for $k_1$ and $k_2$ between the closed-form solution and the MLE. The reason is that for the closed-form solution, camera intrinsic parameters are estimated assuming no distortion, and the predicted outer points lie closer to the image center than the detected ones. The subsequent distortion estimation tries to spread the outer points and increase the scale in order to reduce the distances, although the distortion shape (with positive $k_1$, called pincushion distortion) does not correspond to the real distortion (with negative $k_1$, called barrel distortion). The nonlinear refining (MLE) finally recovers the correct distortion shape. The estimated distortion parameters allow us to correct the distortion in the original images. Figure 5 displays the first two such distortion-corrected images, which should be compared with the first two images shown in Figure 4. We see clearly that the curved pattern in the original images is straightened.

**Variation of the calibration result.** In Table 1, we have shown the calibration results with 2 through 5 images, and

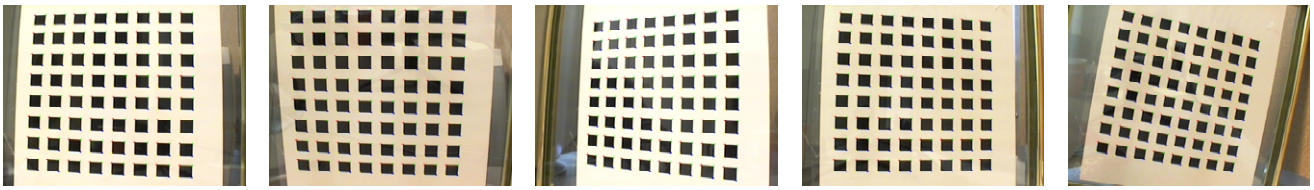

Figure 4: Five images of a model plane, together with the extracted corners (indicated by cross, but too small to be observable)

Table 1: Results with real data of 2 through 5 images

| nb | 2 images | | | 3 images | | | 4 images | | | 5 images | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | initial | final | $\sigma$ | initial | final | $\sigma$ | initial | final | $\sigma$ | initial | final | $\sigma$ |
| $\alpha$ | 825.59 | 830.47 | 4.74 | 917.65 | 830.80 | 2.06 | 876.62 | 831.81 | 1.56 | 877.16 | 832.50 | 1.41 |
| $\beta$ | 825.26 | 830.24 | 4.85 | 920.53 | 830.69 | 2.10 | 876.22 | 831.82 | 1.55 | 876.80 | 832.53 | 1.38 |
| $c$ | 0 | 0 | 0 | 2.2956 | 0.1676 | 0.109 | 0.0658 | 0.2867 | 0.095 | 0.1752 | 0.2045 | 0.078 |
| $u_0$ | 295.79 | 307.03 | 1.37 | 277.09 | 305.77 | 1.45 | 301.31 | 304.53 | 0.86 | 301.04 | 303.96 | 0.71 |
| $v_0$ | 217.69 | 206.55 | 0.93 | 223.36 | 206.42 | 1.00 | 220.06 | 206.79 | 0.78 | 220.41 | 206.56 | 0.66 |
| $k_1$ | 0.161 | $-0.227$ | 0.006 | 0.128 | $-0.229$ | 0.006 | 0.145 | $-0.229$ | 0.005 | 0.136 | $-0.228$ | 0.003 |
| $k_2$ | $-1.955$ | 0.194 | 0.032 | $-1.986$ | 0.196 | 0.034 | $-2.089$ | 0.195 | 0.028 | $-2.042$ | 0.190 | 0.025 |
| RMS | 0.761 | 0.295 | | 0.987 | 0.393 | | 0.927 | 0.361 | | 0.881 | 0.335 | |

Table 2: Variation of the calibration results among all quadruples of images

| quadruple | (1234) | (1235) | (1245) | (1345) | (2345) | mean | deviation |
|---|---|---|---|---|---|---|---|
| $\alpha$ | 831.81 | 832.09 | 837.53 | 829.69 | 833.14 | 832.85 | 2.90 |
| $\beta$ | 831.82 | 832.10 | 837.53 | 829.91 | 833.11 | 832.90 | 2.84 |
| $c$ | 0.2867 | 0.1069 | 0.0611 | 0.1363 | 0.1096 | 0.1401 | 0.086 |
| $u_0$ | 304.53 | 304.32 | 304.57 | 303.95 | 303.53 | 304.18 | 0.44 |
| $v_0$ | 206.79 | 206.23 | 207.30 | 207.16 | 206.33 | 206.76 | 0.48 |
| $k_1$ | $-0.229$ | $-0.228$ | $-0.230$ | $-0.227$ | $-0.229$ | $-0.229$ | 0.001 |
| $k_2$ | 0.195 | 0.191 | 0.193 | 0.179 | 0.190 | 0.190 | 0.006 |
| RMS | 0.361 | 0.357 | 0.262 | 0.358 | 0.334 | 0.334 | 0.04 |

we have found that the results are very consistent with each other. In order to further investigate the stability of the proposed algorithm, we have applied it to all combinations of 4 images from the available 5 images. The results are shown in Table 2, where the third column (1235), for example, displays the result with the quadruple of the first, second, third, and fifth image. The last two columns display the mean and sample deviation of the five sets of results. The sample deviations for all parameters are quite small, which implies that the proposed algorithm is quite stable. The value of the skew parameter $c$ is not significant from 0, since the coefficient of variation, 0.086/0.1401 = 0.6, is large. Indeed, $c = 0.1401$ with $\alpha = 832.85$ corresponds to 89.99 degrees, very close to 90 degrees, for the angle between the two image axes. We have also computed the aspect ratio $\alpha/\beta$ for each quadruple. The mean of the aspect ratio is equal to 0.99995 with sample deviation 0.00012. It is therefore very close to 1, i.e., the pixels are square.

**Application to image-based modeling.** Two images of a tea tin (see Fig. 6) were taken by the same camera as used above for calibration. Mainly two sides are visible. We manually picked 8 point matches on each side, and the structure-from-motion software we developed earlier was run on these 16 point matches to build a partial model of the tea tin. The reconstructed model is in VRML, and three rendered views are shown in Fig. 7. The reconstructed points on each side are indeed coplanar, and we computed the angle between the two reconstructed planes which is 94.7°. Although we do not have the ground truth, but the two sides of the tea tin are indeed almost orthogonal to each other.


Figure 6: Two images of a tea tin

Figure 7: Three rendered views of the reconstructed tea tin

All the real data and results together with the software are available from the following Web page:
http://research.microsoft.com/~zhang/Calib/

## 6. Conclusion

In this paper, we have developed a new flexible technique calibrate to easily a camera. The technique only requires the camera to observe a planar pattern from a few (at least two) different orientations. We can move either the camera or the planar pattern. The motion does not need to be known. Radial lens distortion is modeled. The proposed procedure consists of a closed-form solution, followed by a nonlinear refining based on maximum likelihood criterion. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques which use expensive equipment such as two or three orthogonal planes, the proposed technique gains considerable flexibility.

**Acknowledgment.** Thanks go to Brian Guenter for his software of corner extraction and for many discussions, and to Bill Triggs for insightful comments. Thanks go to Andrew Zisserman for bringing his CVPR98 work [13] to my attention, which uses the same constraint but in different form. Thanks go to Bill Triggs and Gideon Stein for suggesting experiments on model imprecision, which can be found in the technical report [24]. Anandan and Charles Loop have checked the English of an early version.

## A. Estimation of the Homography Between the Model Plane and its Image

There are many ways to estimate the homography between the model plane and its image. Here, we present a technique based on maximum likelihood criterion. Let $\mathtt{M}_i$ and $\mathbf{m}_i$ be the model and image points, respectively. Ideally, they should satisfy (2). In practice, they don't because of noise in the extracted image points. Let's assume that $\mathbf{m}_i$ is corrupted by Gaussian noise with mean $\mathbf{0}$ and covariance matrix $\Lambda_{\mathbf{m}_i}$. Then, the maximum likelihood estimation of $\mathbf{H}$ is obtained by minimizing the following functional

$$\sum_i (\mathbf{m}_i - \hat{\mathbf{m}}_i)^T \Lambda_{\mathbf{m}_i}^{-1} (\mathbf{m}_i - \hat{\mathbf{m}}_i) \ ,$$

where $\hat{\mathbf{m}}_i = \frac{1}{\bar{\mathbf{h}}_3^T \tilde{\mathtt{M}}_i} \begin{bmatrix} \bar{\mathbf{h}}_1^T \tilde{\mathtt{M}}_i \\ \bar{\mathbf{h}}_2^T \tilde{\mathtt{M}}_i \end{bmatrix}$ with $\bar{\mathbf{h}}_i$, the $i^{\text{th}}$ row of $\mathbf{H}$.

In practice, we simply assume $\Lambda_{\mathbf{m}_i} = \sigma^2 \mathbf{I}$ for all $i$. This is reasonable if points are extracted independently with the same procedure. In this case, the above problem becomes a nonlinear least-squares one, i.e., $\min_{\mathbf{H}} \sum_i \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|^2$. The nonlinear minimization is conducted with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [16]. This requires an initial guess, which can be obtained as follows.

Let $\mathbf{x} = [\bar{\mathbf{h}}_1^T, \bar{\mathbf{h}}_2^T, \bar{\mathbf{h}}_3^T]^T$. Then equation (2) can be rewritten as

$$\begin{bmatrix} \tilde{\mathtt{M}}^T & \mathbf{0}^T & -u\tilde{\mathtt{M}}^T \\ \mathbf{0}^T & \tilde{\mathtt{M}}^T & -v\tilde{\mathtt{M}}^T \end{bmatrix} \mathbf{x} = \mathbf{0} \ .$$

When we are given $n$ points, we have $n$ above equations, which can be written in matrix equation as $\mathbf{L}\mathbf{x} = \mathbf{0}$, where $\mathbf{L}$ is a $2n \times 9$ matrix. As $\mathbf{x}$ is defined up to a scale factor, the solution is well known to be the right singular vector of $\mathbf{L}$ associated with the smallest singular value (or equivalently, the eigenvector of $\mathbf{L}^T\mathbf{L}$ associated with the smallest eigenvalue).

In $\mathbf{L}$, some elements are constant 1, some are in pixels, some are in world coordinates, and some are multiplication of both. This makes $\mathbf{L}$ poorly conditioned numerically. Much better results can be obtained by performing a simple data normalization, such as the one proposed in [11], prior to running the above procedure.

## B. Extraction of the Intrinsic Parameters from Matrix B

The matrix $\mathbf{B}$, as described in Sect. 3.1, is estimated up to a scale factor, i.e., , $\mathbf{B} = \lambda \mathbf{A}^{-T} \mathbf{A}$ with $\lambda$ an arbitrary scale. Without difficulty, we can uniquely extract the intrinsic pa-

rameters from matrix $\mathbf{B}$.

$$
\begin{aligned}
v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\
\lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\
\alpha &= \sqrt{\lambda/B_{11}} \\
\beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\
c &= -B_{12}\alpha^2\beta/\lambda \\
u_0 &= cv_0/\alpha - B_{13}\alpha^2/\lambda \ .
\end{aligned}
$$

## C. Approximating a $3 \times 3$ matrix by a Rotation Matrix

The problem considered in this section is to solve the best rotation matrix $\mathbf{R}$ to approximate a given $3 \times 3$ matrix $\mathbf{Q}$. Here, "best" is in the sense of the smallest Frobenius norm of the difference $\mathbf{R} - \mathbf{Q}$. The solution can be found in our technical report [24].

## References


[1] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *Proc. 6th International Conference on Computer Vision*, pages 790–796, Jan. 1998.

[2] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[3] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *The International Journal of Computer Vision*, 4(2):127–140, Mar. 1990.

[4] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, 1975.

[5] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[6] O. Faugeras, T. Luong, and S. Maybank. Camera self-calibration: theory and experiments. In *Proc. 2nd ECCV*, pages 321–334, May 1992.

[7] O. Faugeras and G. Toscani. The calibration problem for stereo. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, FL, June 1986.

[8] S. Ganapathy. Decomposition of transformation matrices for robot vision. *Pattern Recognition Letters*, 2:401–412, Dec. 1984.

[9] D. Gennery. Stereo-camera calibration. In *Proc. 10th Image Understanding Workshop*, pages 101–108, 1979.

[10] R. Hartley. Self-calibration from multiple views with a rotating camera. In *Proc. 3rd European Conference on Computer Vision*, pages 471–478, Stockholm, Sweden, May 1994.

[11] R. Hartley. In defence of the 8-point algorithm. In *Proc. 5th International Conference on Computer Vision*, pages 1064–1070, Boston, MA, June 1995.

[12] R. I. Hartley. An algorithm for self calibration from several views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 908–912, Seattle, WA, June 1994.

[13] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, Santa Barbara, California, June 1998.

[14] Q.-T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *The International Journal of Computer Vision*, 22(3):261–289, 1997.

[15] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *The International Journal of Computer Vision*, 8(2):123–152, Aug. 1992.

[16] J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag, 1977.

[17] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, 4th ed., 1980.

[18] G. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. In *Proc. 5th International Conference on Computer Vision*, pages 230–236, Cambridge, Massachusetts, June 1995.

[19] B. Triggs. Autocalibration from planar scenes. In *Proc. 5th European Conference on Computer Vision*, pages 89–105, Freiburg, Germany, June 1998.

[20] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, Aug. 1987.

[21] G. Wei and S. Ma. A complete two-plane camera calibration method and experimental comparisons. In *Proc. Fourth International Conference on Computer Vision*, pages 439–446, Berlin, May 1993.

[22] G. Wei and S. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, 1994.

[23] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct. 1992.

[24] Z. Zhang. *A Flexible New Technique for Camera Calibration*. Technical Report MSR-TR-98-71, Microsoft Research, December 1998. Available together with the software at http://research.microsoft.com/~zhang/Calib/