Camera Calibration with One-Dimensional Objects

Zhengyou Zhang

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

Abstract. Camera calibration has been studied extensively in computer vision and photogrammetry, and the proposed techniques in the literature include those using 3D apparatus (two or three planes orthogonal to each other, or a plane undergoing a pure translation, etc.), 2D objects (planar patterns undergoing unknown motions), and 0D features (self-calibration using unknown scene points). This paper yet proposes a new calibration technique using 1D objects (points aligned on a line), thus filling the missing dimension in calibration. In particular, we show that camera calibration is not possible with free-moving 1D objects, but can be solved if one point is fixed. A closed-form solution is developed if six or more observations of such a 1D object are made. For higher accuracy, a nonlinear technique based on the maximum likelihood criterion is then used to refine the estimate. Besides the theoretical aspect, the proposed technique is also important in practice especially when calibrating multiple cameras mounted apart from each other, where the calibration objects are required to be visible simultaneously.

Keywords: Camera calibration

1 Introduction

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 [1,3] to cite a few), and more recently in computer vision ([8,7,19,6,21,20,14,5] to cite a few). According to the dimension of the calibration objects, we can classify those techniques roughly into three categories.

3D reference object based calibration. Camera 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 [4]. 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 [19], which equivalently provides 3D reference points. This approach requires an expensive calibration apparatus and an elaborate setup.

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2353, pp. 161–174, 2002.

2D plane based calibration. Techniques in this category requires to observe a planar pattern shown at a few different orientations [22,17]. Different from Tsai's technique [19], the knowledge of the plane motion is not necessary. Because almost anyone can make such a calibration pattern by him/her-self, the setup is easier for camera calibration.

Self-calibration. Techniques in this category do not use any calibration object, and can be considered as 0D approach because only image point correspondences are required. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints [14,13] 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 [12,10]. Although no calibration objects are necessary, a large number of parameters need to be estimated, resulting in a much harder mathematical problem.

Other techniques exist: vanishing points for orthogonal directions [2,11], and calibration from pure rotation [9,16].

To our knowledge, there does not exist any calibration technique reported in the literature which uses one-dimensional (1D) calibration objects, and this is the topic we will investigate in this paper. In particular, we will consider 1D objects composed of a set of collinear points. Unlike techniques using 3D reference objects, other techniques requires taking several snapshots of calibration objects or the environment. This is the price we pay, although insignificant in practice, by using poorer knowledge of the observation. This is also the case with calibration using 1D objects.

Besides the theoretical aspect of using 1D objects in camera calibration, it is also very important in practice especially when multi-cameras are involved in the environment. To calibrate the relative geometry between multiple cameras, it is necessary for all involving cameras to simultaneously observe a number of points. It is hardly possible to achieve this with 3D or 2D calibration apparatus[1] if one camera is mounted in the front of a room while another in the back. This is not a problem for 1D objects. We can for example use a string of balls hanging from the ceiling.

The paper is organized as follows. Section 2 examines possible setups with 1D objects for camera calibration. Section 3 describes in detail how to solve camera calibration with 1D objects. Both closed-form solution and nonlinear minimization based on maximum likelihood criterion are proposed. Section 4 provides experimental results with both computer simulated data and real images. Finally, Section 5 concludes the paper with perspective of this work.

[1] An exception is when those apparatus are made transparent; then the cost would be much higher.

2 Preliminaries

We examine possible setups with 1D objects for camera calibration. 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 $\mathbf{m}$ (perspective projection) is given by

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

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, and $\mathbf{A}$ is called the camera intrinsic matrix, with (u_0, v_0) the coordinates of the principal point, α and β the scale factors in image u and v axes, and γ the parameter describing the skew of the two image axes. The task of camera calibration is to determine these five intrinsic parameters.

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

2.2 Setups with Free-Moving 1D Calibration Objects

We now examine possible setups with 1D objects for camera calibration. As already mentioned in the introduction, we need to have several observations of the 1D objects. Without loss of generality, we choose the camera coordinate system to define the 1D objects; therefore, $\mathbf{R} = \mathbf{I}$ and $\mathbf{t} = \mathbf{0}$ in (1).

Two points with known distance. This could be the two endpoints of a stick, and we take a number of images while waving freely the stick. Let $\mathtt{A}$ and $\mathtt{B}$ be the two 3D points, and $\mathbf{a}$ and $\mathbf{b}$ be the observed image points. Because the distance between $\mathtt{A}$ and $\mathtt{B}$ is known, we only need 5 parameters to define $\mathtt{A}$ and $\mathtt{B}$. For example, we need 3 parameters to specify the coordinates of $\mathtt{A}$ in the camera coordinate system, and 2 parameters to define the orientation of the line $\mathtt{AB}$. On the other hand, each image point provides two equations according to (1), giving in total 4 equations. Given N observations of the stick, we have 5 intrinsic parameters and $5N$ parameters for the point positions to estimate, i.e., the total number of unknowns is $5 + 5N$. However, we only have $4N$ equations. Camera calibration is thus impossible.

Three collinear points with known distances. By adding an additional point, say C, the number of unknowns for the point positions still remains the same, i.e., $5 + 5N$, because of known distances of C to A and B. For each observation, we have three image points, yielding in total $6N$ equations. Calibration seems to be plausible, but is in fact not. This is because the three image points for each observation must be collinear. Collinearity is preserved by perspective projection. We therefore only have 5 independent equations for each observation. The total number of independent equations, $5N$, is always smaller than the number of unknowns. Camera calibration is still impossible.

Four or more collinear points with known distances. As seen above, when the number of points increases from two to three, the number of independent equations (constraints) increases by one for each observation. If we have a fourth point, will we have in total $6N$ independent equations? If so, we would be able to solve the problem because the number of unknowns remains the same, i.e., $5 + 5N$, and we would have more than enough constraints if $N \geq 5$. The reality is that the addition of the fourth point or even more points does not increase the number of independent equations. It will always be $5N$ for any four or more collinear points. This is because the cross ratio is preserved under perspective projection. With known cross ratios and three collinear points, whether they are in space or in images, other points are determined exactly.

2.3 Setups with 1D Calibration Objects Moving Around a Fixed Point

From the above discussion, calibration is impossible with a free moving 1D calibration object, no matter how many points on the object. Now let us examine what happens if one point is fixed. In the sequel, without loss of generality, point A is the fixed point, and $\mathbf{a}$ is the corresponding image point. We need 3 parameters, which are unknown, to specify the coordinates of A in the camera coordinate system, while image point $\mathbf{a}$ provides two scalar equations according to (1).

Two points with known distance. They could be the endpoints of a stick, and we move the stick around the endpoint that is fixed. Let B be the free endpoint and $\mathbf{b}$, its corresponding image point. For each observation, we need 2 parameters to define the orientation of the line AB and therefore the position of B because the distance between A and B is known. Given N observations of the stick, we have 5 intrinsic parameters, 3 parameters for A and $2N$ parameters for the free endpoint positions to estimate, i.e., the total number of unknowns is $8 + 2N$. However, each observation of $\mathbf{b}$ provides two equations, so together with $\mathbf{a}$ we only have in total $2 + 2N$ equations. Camera calibration is thus impossible.

Three collinear points with known distances. As already explained in the last subsection, by adding an additional point, say C, the number of unknowns for the point positions still remains the same, i.e., $8 + 2N$. For each observation, $\mathbf{b}$

provides two equations, but **c** only provides one additional equation because of the collinearity of **a**, **b** and **c**. Thus, the total number of equations is $2 + 3N$ for N observations. By counting the numbers, we see that if we have 6 or more observations, we should be able to solve camera calibration, and this is the case as we shall show in the next section.

Four or more collinear points with known distances. Again, as already explained in the last subsection, The number of unknowns and the number of independent equations remain the same because of invariance of cross-ratios. This said, the more collinear points we have, the more accurate camera calibration will be in practice because data redundancy can combat the noise in image data.

3 Solving Camera Calibration with 1D Objects

In this section, we describe in detail how to solve the camera calibration problem from a number of observations of a 1D object consisting of 3 collinear points moving around one of them. We only consider this minimal configuration, but it is straightforward to extend the result if a calibration object has four or more collinear points.

3.1 Basic Equations

Refer to Figure 1. Point $\mathtt{A}$ is the fixed point in space, and the stick AB moves around $\mathtt{A}$. The length of the stick AB is known to be L, i.e.,

$$\|\mathtt{B} - \mathtt{A}\| = L \; . \tag{2}$$

The position of point $\mathtt{C}$ is also known with respect to $\mathtt{A}$ and $\mathtt{B}$, and therefore

$$\mathtt{C} = \lambda_A \mathtt{A} + \lambda_B \mathtt{B} \; , \tag{3}$$

where λ_A and λ_B are known. If $\mathtt{C}$ is the midpoint of AB, then $\lambda_A = \lambda_B = 0.5$. Points **a**, **b** and **c** on the image plane are projection of space points $\mathtt{A}$, $\mathtt{B}$ and $\mathtt{C}$, respectively.

Without loss of generality, we choose the camera coordinate system to define the 1D objects; therefore, $\mathbf{R} = \mathbf{I}$ and $\mathbf{t} = \mathbf{0}$ in (1). Let the unknown depths for $\mathtt{A}$, $\mathtt{B}$ and $\mathtt{C}$ be z_A, z_B and z_C, respectively. According to (1), we have

$$\mathtt{A} = z_A \mathbf{A}^{-1} \widetilde{\mathbf{a}} \tag{4}$$

$$\mathtt{B} = z_B \mathbf{A}^{-1} \widetilde{\mathbf{b}} \tag{5}$$

$$\mathtt{C} = z_C \mathbf{A}^{-1} \widetilde{\mathbf{c}} \; . \tag{6}$$

Substituting them into (3) yields

$$z_C \widetilde{\mathbf{c}} = z_A \lambda_A \widetilde{\mathbf{a}} + z_B \lambda_B \widetilde{\mathbf{b}} \tag{7}$$

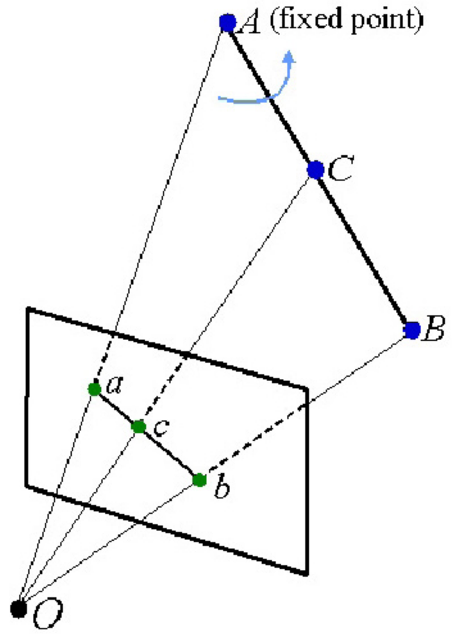


Fig. 1. Illustration of 1D calibration objects

after eliminating $\mathbf{A}^{-1}$ from both sides. By performing cross-product on both sides of the above equation with $\widetilde{\mathbf{c}}$, we have

$$z_A \lambda_A (\widetilde{\mathbf{a}} \times \widetilde{\mathbf{c}}) + z_B \lambda_B (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}}) = \mathbf{0} .$$

In turn, we obtain

$$z_B = -z_A \frac{\lambda_A (\widetilde{\mathbf{a}} \times \widetilde{\mathbf{c}}) \cdot (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}})}{\lambda_B (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}}) \cdot (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}})} . \tag{8}$$

From (2), we have

$$\|\mathbf{A}^{-1}(z_B \widetilde{\mathbf{b}} - z_A \widetilde{\mathbf{a}})\| = L .$$

Substituting z_B by (8) gives

$$z_A \|\mathbf{A}^{-1}\big(\widetilde{\mathbf{a}} + \frac{\lambda_A (\widetilde{\mathbf{a}} \times \widetilde{\mathbf{c}}) \cdot (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}})}{\lambda_B (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}}) \cdot (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}})} \widetilde{\mathbf{b}}\big)\| = L .$$

This is equivalent to

$$z_A^2 \mathbf{h}^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h} = L^2 \tag{9}$$

with

$$\mathbf{h} = \widetilde{\mathbf{a}} + \frac{\lambda_A (\widetilde{\mathbf{a}} \times \widetilde{\mathbf{c}}) \cdot (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}})}{\lambda_B (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}}) \cdot (\widetilde{\mathbf{b}} \times \widetilde{\mathbf{c}})} \widetilde{\mathbf{b}} \,. \tag{10}$$

Equation (9) contains the unknown intrinsic parameters $\mathbf{A}$ and the unknown depth, z_A, of the fixed point $\mathtt{A}$. It is the basic constraint for camera calibration with 1D objects. Vector $\mathbf{h}$, given by (10), can be computed from image points and known λ_A and λ_B. Since the total number of unknowns is 6, we need at least six observations of the 1D object for calibration. Note that $\mathbf{A}^{-T}\mathbf{A}$ actually describes the image of the absolute conic [12].

3.2 Closed-Form Solution

Let

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

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} . \tag{12}$$

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

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

Let $\mathbf{h} = [h_1, h_2, h_3]^T$, and $\mathbf{x} = z_A^2 \mathbf{b}$, then equation (9) becomes

$$\mathbf{v}^T \mathbf{x} = L^2 \tag{14}$$

with

$$\mathbf{v} = [h_1^2, 2h_1h_2, h_2^2, 2h_1h_3, 2h_2h_3, h_3^2]^T \,.$$

When N images of the 1D object are observed, by stacking n such equations as (14) we have

$$\mathbf{V}\mathbf{x} = L^2 \mathbf{1} \,, \tag{15}$$

where $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_N]^T$ and $\mathbf{1} = [1, \dots, 1]^T$. The least-squares solution is then given by

$$\mathbf{x} = L^2 (\mathbf{V}^T\mathbf{V})^{-1} \mathbf{V}^T \mathbf{1} \,. \tag{16}$$

Once $\mathbf{x}$ is estimated, we can compute all the unknowns based on $\mathbf{x} = z_A^2 \mathbf{b}$. Let $\mathbf{x} = [x_1, x_2, \dots, x_6]^T$. Without difficulty, we can uniquely extract the intrinsic parameters and the depth z_A as

$$
\begin{aligned}
v_0 &= (x_2 x_4 - x_1 x_5)/(x_1 x_3 - x_2^2) \\
z_A &= \sqrt{x_6 - [x_4^2 + v_0(x_2 x_4 - x_1 x_5)]/x_1} \\
\alpha &= \sqrt{z_A/x_1} \\
\beta &= \sqrt{z_A x_1/(x_1 x_3 - x_2^2)} \\
\gamma &= -x_2 \alpha^2 \beta / z_A \\
u_0 &= \gamma v_0/\alpha - x_4 \alpha^2 / z_A \; .
\end{aligned}
$$

At this point, we can compute z_B according to (8), so points A and B can be computed from (4) and (5), while point C can be computed according to (3).

3.3 Nonlinear Optimization

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 the 1D calibration object and there are 3 points on the object. Point A is fixed, and points B and C moves around A. 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:

$$
\sum_{i=1}^{N} \left(\|\mathbf{a}_i - \boldsymbol{\phi}(\mathbf{A}, \mathtt{A})\|^2 + \|\mathbf{b}_i - \boldsymbol{\phi}(\mathbf{A}, \mathtt{B}_i)\|^2 + \|\mathbf{c}_i - \boldsymbol{\phi}(\mathbf{A}, \mathtt{C}_i)\|^2 \right) \; , \tag{17}
$$

where $\boldsymbol{\phi}(\mathbf{A}, \mathtt{M})$ ($\mathtt{M} \in \{\mathtt{A}, \mathtt{B}_i, \mathtt{C}_i\}$) is the projection of point M onto the image, according to equations (4) to (6). More precisely, $\boldsymbol{\phi}(\mathbf{A}, \mathtt{M}) = \frac{1}{z_M} \mathbf{A}\mathtt{M}$, where z_M is the z-component of M.

The unknowns to be estimated are:

– 5 camera intrinsic parameters α, β, γ, u_0 and v_0 that define matrix $\mathbf{A}$;
– 3 parameters for the coordinates of the fixed point A;
– $2N$ additional parameters to define points $\mathtt{B}_i$ and $\mathtt{C}_i$ at each instant (see below for more details).

Therefore, we have in total $8 + 2N$ unknowns. Regarding the parameterization for B and C, we use the spherical coordinates ϕ and θ to define the direction of the 1D calibration object, and point B is then given by

$$
\mathtt{B} = \mathtt{A} + L \begin{bmatrix} \sin\theta \cos\phi \\ \sin\theta \sin\phi \\ \cos\theta \end{bmatrix}
$$

where L is the known distance between A and B. In turn, point C is computed according to (3). We therefore only need 2 additional parameters for each observation.

Minimizing (17) is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [15]. It requires an initial guess of $\mathbf{A}, \mathtt{A}, \{\mathtt{B}_i, \mathtt{C}_i | i = 1..N\}$ which can be obtained using the technique described in the last subsection.

4 Experimental Results

The proposed algorithm has been tested on both computer simulated data and real data.

4.1 Computer Simulations

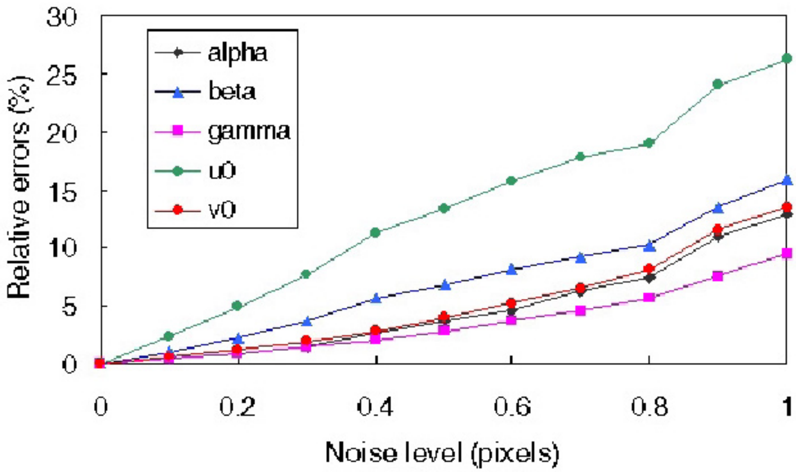
The simulated camera has the following property: $\alpha = 1000$, $\beta = 1000$, $\gamma = 0$, $u_0 = 320$, and $v_0 = 240$. The image resolution is 640×480. A stick of 70 cm is simulated with the fixed point $\mathtt{A}$ at $[0, 35, 150]^T$. The other endpoint of the stick is $\mathtt{B}$, and $\mathtt{C}$ is located at the half way between $\mathtt{A}$ and $\mathtt{B}$. We have generated 100 random orientations of the stick by sampling θ in $[\pi/6, 5\pi/6]$ and ϕ in $[\pi, 2\pi]$ according to uniform distribution. Points $\mathtt{A}$, $\mathtt{B}$, and $\mathtt{C}$ are then projected onto the image.

Gaussian noise with 0 mean and σ standard deviation is added to the projected image points $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$. The estimated camera parameters are compared with the ground truth, and we measure their relative errors with respect to the focal length α. Note that we measure the relative errors in (u_0, v_0) with respect to α, as proposed by Triggs in [18]. He pointed out that the absolute errors in (u_0, v_0) is not geometrically meaningful, while computing the relative error is equivalent to measuring the angle between the true optical axis and the estimated one.

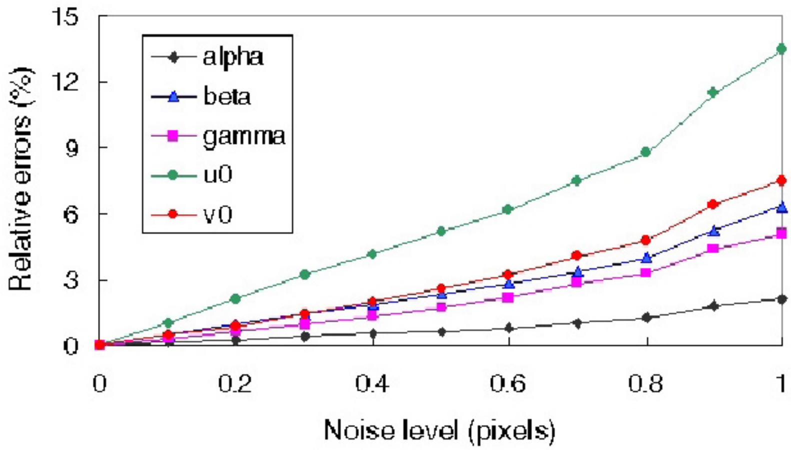
We vary the noise level from 0.1 pixels to 1 pixel. For each noise level, we perform 120 independent trials, and the results shown in Fig. 2 are the average. Figure 2a displays the relative errors of the closed-form solution while Figure 2b displays those of the nonlinear minimization result. Errors increase almost linearly with the noise level. The nonlinear minimization refines the closed-form solution, and produces significantly better result (with 50% less errors). At 1 pixel noise level, the errors for the closed-form solution are about 12%, while those for the nonlinear minimization are about 6%.

Table 1. Calibration results with real data.

Solution	α	β	γ	u_0	v_0
Closed-form	889.49	818.59	-0.1651 (90.01°)	297.47	234.33
Nonlinear	838.49	799.36	4.1921 (89.72°)	286.74	219.89
Plane-based	828.92	813.33	-0.0903 (90.01°)	305.23	235.17
Relative difference	1.15%	1.69%	0.52% (0.29°)	2.23%	1.84%



(a) Closed-form solution



(b) Nonlinear optimization

Fig. 2. Calibration errors with respect to the noise level of the image points.

4.2 Real Data

For the experiment with real data, I used three toy beads from my kids and strung them together with a stick. The beads are approximately 14 cm apart (i.e., $L = 28$). I then moves the stick around while trying to fix one end with the aid of a book. A video of 150 frames was recorded, and four sample images

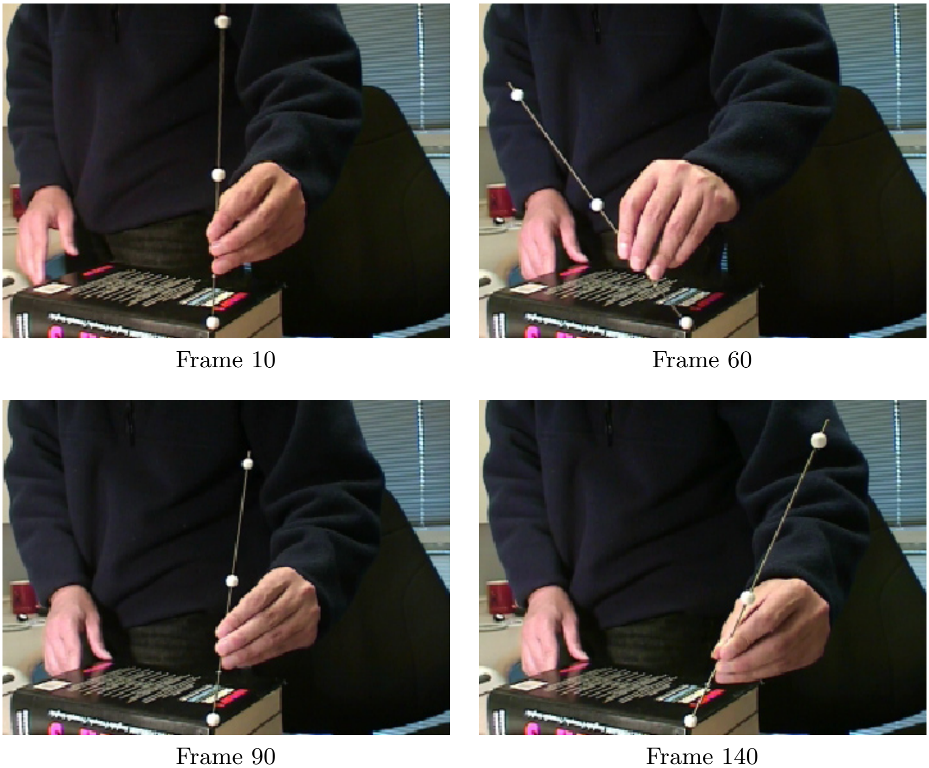


Fig. 3. Sample images of a 1D object used for camera calibration.

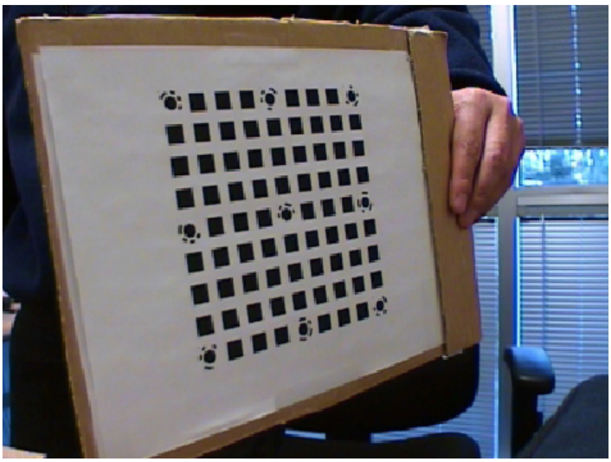


Fig. 4. A sample image of the planar pattern used for camera calibration.

are shown in Fig. 3. A bead in the image is modeled as a Gaussian blob in the RGB space, and the centroid of each detected blob is the image point we use for camera calibration. The proposed algorithm is therefore applied to the 150 observations of the beads, and the estimated camera parameters are provided in Table 1. The first row is the estimation from the closed-form solution, while the second row is the refined result after nonlinear minimization. For the image skew parameter γ, we also provide the angle between the image axes in parenthesis (it should be very close to $90°$).

For comparison, we also used the plane-based calibration technique described in [22] to calibrate the same camera. Five images of a planar pattern were taken, and one of them is shown in Fig. 4. The calibration result is shown in the third row of Table 1. The fourth row displays the relative difference between the plane-based result and the nonlinear solution with respect to the focal length (we use 828.92). As we can observe, the difference is about 2%.

There are several sources contributing to this difference. Besides obviously the image noise and imprecision of the extracted data points, one source is our current rudimentary experimental setup:

- The supposed-to-be fixed point was not fixed. It slipped around on the surface.
- The positioning of the beads was done with a ruler using eye inspection.

Considering all the factors, the proposed algorithm is very encouraging.

5 Conclusion

In this paper, we have investigated the possibility of camera calibration using one-dimensional objects. One-dimensional calibration objects consist of three or more collinear points with known relative positioning. In particular, we have shown that camera calibration is not possible with free-moving 1D objects, but can be solved if one point is fixed. A closed-form solution has been developed if six or more observations of such a 1D object are made. For higher accuracy, a nonlinear technique based on the maximum likelihood criterion is used to refine the estimate. Both computer simulation and real data have been used to test the proposed algorithm, and very encouraging results have been obtained.

Camera calibration has been studied extensively in computer vision and photogrammetry, and the proposed techniques in the literature include those using 3D apparatus (two or three planes orthogonal to each other, or a plane undergoing a pure translation, etc.), 2D objects (planar patterns undergoing unknown motions), and 0D features (self-calibration using unknown scene points). This proposed calibration technique uses 1D objects (points aligned on a line), thus filling the missing dimension in calibration. Besides the theoretical aspect, the proposed technique is also important in practice especially when calibrating multiple cameras mounted apart from each other, where the calibration objects are required to be visible simultaneously.

Currently, we are planning to work on the following two problems:

- This paper has only examined the minimal configuration, that is, 1D object with three points. With four or more points on a line, although we do not gain any theoretical constraints, we should be able to obtain more accurate calibration results because of data redundancy in combating noise in image points.
- The proposed algorithm assumes that the fixed point is visible by the camera. It would be more flexible for camera calibration if the fixed point could be invisible. In that case, we can for example hang a string of small balls from the ceiling, and calibrate multiple cameras in the room by swinging the string.

For the second point, as suggested by one of the reviewers, the solution is relatively straightforward. The fixed point can be estimated by intersecting lines from different images. Alternatively, if more than three points are visible in each image, the fixed point can be determined from the known cross ratio.

Acknowledgment. The author thanks one of the reviewers for suggestion of a solution to the case where the fixed point is not observed by the camera.

References

1. Duane C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
2. Bruno Caprile and Vincent Torre. Using Vanishing Points for Camera Calibration. *The International Journal of Computer Vision*, 4(2):127–140, March 1990.
3. W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, 1975.
4. Olivier Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint.* MIT Press, 1993.
5. Olivier Faugeras, Tuan Luong, and Steven Maybank. Camera self-calibration: theory and experiments. In G. Sandini, editor, *Proc 2nd ECCV*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
6. Olivier Faugeras and Giorgio Toscani. The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, FL, June 1986. IEEE.
7. S. Ganapathy. Decomposition of transformation matrices for robot vision. *Pattern Recognition Letters*, 2:401–412, December 1984.
8. D. Gennery. Stereo-camera calibration. In *Proceedings of the 10th Image Understanding Workshop*, pages 101–108, 1979.
9. Richard Hartley. Self-calibration from multiple views with a rotating camera. In J-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800-801 of *Lecture Notes in Computer Science*, pages 471–478, Stockholm, Sweden, May 1994. Springer-Verlag.
10. Richard I. Hartley. An algorithm for self calibration from several views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 908–912, Seattle, WA, June 1994. IEEE.

11. D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, Santa Barbara, California, June 1998. IEEE Computer Society.
12. Q.-T. Luong and O.D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *The International Journal of Computer Vision*, 22(3):261–289, 1997.
13. Quang-Tuan Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systèmes robotiques.* PhD thesis, Université de Paris-Sud, Centre d'Orsay, December 1992.
14. 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, August 1992.
15. J.J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag, 1977.
16. G. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. In *Proc. Fifth International Conference on Computer Vision*, pages 230–236, Cambridge, Massachusetts, June 1995.
17. P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 432–437, Fort Collins, Colorado, June 1999. IEEE Computer Society Press.
18. B. Triggs. Autocalibration from planar scenes. In *Proceedings of the 5th European Conference on Computer Vision*, pages 89–105, Freiburg, Germany, June 1998.
19. Roger 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, August 1987.
20. G.Q. Wei and S.D. 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.
21. 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, October 1992.
22. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.