

Camera Calibration with One-Dimensional Objects

Zhengyou Zhang, *Senior Member, IEEE*

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). Yet, this paper 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. Singularities have also been studied. 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.

Index Terms—Camera calibration, calibration taxonomy, calibration apparatus, 1D objects, singularity, degenerate configuration.

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], [20], [6], [22], [21], [15], [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 3D 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 [20], which equivalently provides 3D reference points. This approach requires an expensive calibration apparatus and an elaborate setup.

2D plane-based calibration. Techniques in this category are required to observe a planar pattern shown at a few different orientations [23], [18]. Different from Tsai's technique [20], the knowledge of the plane motion is not necessary. Because almost anyone can make such a calibration pattern by him/herself, 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 [15], [14] 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 3D structure up to a similarity [13], [10]. Although no calibration objects are necessary, a large number of parameters need to be estimated, resulting in a much harder mathematical problem. A recent overview of this area can be found in [11].

Other techniques exist: vanishing points for orthogonal directions [2], [12] and calibration from pure rotation [9], [17].

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 multicameras 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 apparatus¹ 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

• The author is with Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-mail: zhang@microsoft.com.

Manuscript received 18 Nov. 2002; revised 7 Dec. 2003; accepted 9 Dec. 2003. Recommended for acceptance by L. Quan.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117790.

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

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.

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

$$s\tilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \quad \mathbf{t}]\tilde{\mathbf{M}}, \quad \text{with } \mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

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 Section 1, 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 \mathbf{A} and \mathbf{B} be the two 3D points and \mathbf{a} and \mathbf{b} be the observed image points. Because the distance between \mathbf{A} and \mathbf{B} is known, we only need five parameters to define \mathbf{A} and \mathbf{B} . For example, we need three parameters to specify the coordinates of \mathbf{A} in the camera coordinate system and two parameters to define the orientation of the line \mathbf{AB} . On the other hand, each image point provides two equations according to (1), giving in total four equations. Given N observations of the stick, we have five 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 \mathbf{C} , the number of unknowns for the point positions still remains the same, i.e., $5 + 5N$, because of known distances of \mathbf{C} to \mathbf{A} and \mathbf{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 five 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 \mathbf{A} is the fixed point and \mathbf{a} is the corresponding image point. We need three parameters, which are unknown, to specify the coordinates of \mathbf{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 \mathbf{B} be the free endpoint and \mathbf{b} , its corresponding image point. For each observation, we need two parameters to define the orientation of the line \mathbf{AB} and, therefore, the position of \mathbf{B} because the distance between \mathbf{A} and \mathbf{B} is known. Given N observations of the stick, we have five intrinsic parameters, three parameters for \mathbf{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 section, by adding an additional point, say \mathbf{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 \mathbf{c} only provides one additional equation because of the collinearity of \mathbf{a} , \mathbf{b} , and \mathbf{c} . Thus, the total number of equations is $2 + 3N$ for N observations. By counting the numbers, we see that if we have six 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 section, 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.

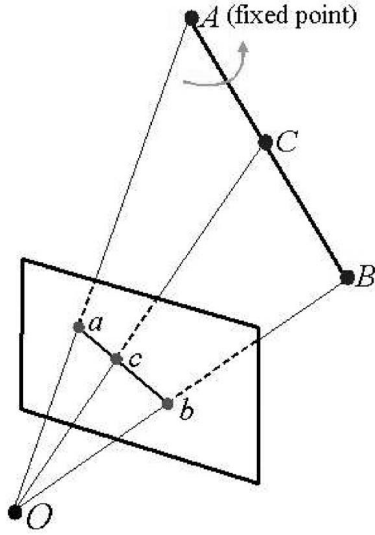


Fig. 1. Illustration of 1D calibration objects.

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 three 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 Fig 1. Point A is the fixed point in space, and the stick AB moves around A. The length of the stick AB is known to be L , i.e.,

$$\|B - A\| = L. \quad (2)$$

The position of point C is also known with respect to A and B and, therefore,

$$C = \lambda_A A + \lambda_B B, \quad (3)$$

where λ_A and λ_B are known. If 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 A, B, and 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 A, B, and C be z_A , z_B , and z_C , respectively. According to (1), we have

$$\mathbf{A} = z_A \mathbf{A}^{-1} \tilde{\mathbf{a}} \quad (4)$$

$$\mathbf{B} = z_B \mathbf{A}^{-1} \tilde{\mathbf{b}} \quad (5)$$

$$\mathbf{C} = z_C \mathbf{A}^{-1} \tilde{\mathbf{c}}. \quad (6)$$

Substituting them into (3) yields

$$z_C \tilde{\mathbf{c}} = z_A \lambda_A \tilde{\mathbf{a}} + z_B \lambda_B \tilde{\mathbf{b}} \quad (7)$$

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

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

In turn, we obtain

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

From (2), we have

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

Substituting z_B by (8) gives

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

This is equivalent to

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

with

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

Equation (9) contains the unknown intrinsic parameters \mathbf{A} and the unknown depth, z_A , of the fixed point 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 six, 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 [13].

3.2 Closed-Form Solution

Let

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

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

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

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

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

$$\mathbf{v}^T \mathbf{x} = L^2 \quad (14)$$

with

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

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

$$\mathbf{V} \mathbf{x} = L^2 \mathbf{1}, \quad (15)$$

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

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

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 three 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:

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

where $\phi(\mathbf{A}, \mathbf{M})$ ($\mathbf{M} \in \{\mathbf{A}, \mathbf{B}_i, \mathbf{C}_i\}$) is the projection of point \mathbf{M} onto the image, according to (4), (5), and (6). More precisely, $\phi(\mathbf{A}, \mathbf{M}) = \frac{1}{z_M} \mathbf{A} \mathbf{M}$, where z_M is the z -component of \mathbf{M} .

The unknowns to be estimated are:

- Five camera intrinsic parameters $\alpha, \beta, \gamma, u_0$, and v_0 that define matrix \mathbf{A} ;
- Three parameters for the coordinates of the fixed point A;
- $2N$ additional parameters to define points \mathbf{B}_i and \mathbf{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

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

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

Minimizing (17) is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in Minpack [16]. It requires an initial guess of $\mathbf{A}, \mathbf{A}, \{\mathbf{B}_i, \mathbf{C}_i | i = 1..N\}$ which can be obtained using the technique described in the last section.

3.4 Estimating the Fixed Point

In the above discussion, we assumed that the image coordinates, \mathbf{a} , of the fixed point A are known. We now describe how to estimate \mathbf{a} by considering whether the fixed point A is visible in the image or not.

Invisible fixed point. The fixed point does not need to be visible in the image. And the camera calibration technique becomes more versatile without the visibility requirement. 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. The fixed point can be estimated by intersecting lines from different images as described below.

Each observation of the 1D object defines an image line. An image line can be represented by a 3D vector $\mathbf{l} = [l_1, l_2, l_3]^T$, defined up to a scale factor such as a point $\mathbf{m} = [u, v]^T$ on the line satisfies $\mathbf{l}^T \tilde{\mathbf{m}} = 0$. In the sequel, we also use (\mathbf{n}, q) to denote line \mathbf{l} , where $\mathbf{n} = [n_1, n_2]^T$ and $q = l_3$. To remove the scale ambiguity, we normalize \mathbf{l} such that $\|\mathbf{l}\| = 1$. Furthermore, each \mathbf{l} is associated with an uncertainty measure represented by a 3×3 covariance matrix Λ .

Given N images of the 1D object, we have N lines: $\{(\mathbf{l}_i, \Lambda_i) | i = 1, \dots, N\}$. Let the fixed point be \mathbf{a} in the image. Obviously, if there is no noise, we have $\mathbf{l}_i^T \tilde{\mathbf{a}} = 0$, or $\mathbf{n}_i^T \mathbf{a} + q_i = 0$. Therefore, we can estimate \mathbf{a} by minimizing

$$\begin{aligned} \mathcal{F} &= \sum_{i=1}^N w_i \|\mathbf{l}_i^T \tilde{\mathbf{a}}\|^2 = \sum_{i=1}^N w_i \|\mathbf{n}_i^T \mathbf{a} + q_i\|^2 \\ &= \sum_{i=1}^N w_i (\mathbf{a}^T \mathbf{n}_i \mathbf{n}_i^T \mathbf{a} + 2q_i \mathbf{n}_i^T \mathbf{a} + q_i^2), \end{aligned} \quad (18)$$

where w_i is a weighting factor (see below). By setting the derivative of \mathcal{F} with respect to \mathbf{a} to 0, we obtain the solution, which is given by

$$\mathbf{a} = - \left(\sum_{i=1}^N w_i \mathbf{n}_i \mathbf{n}_i^T \right)^{-1} \left(\sum_{i=1}^N w_i q_i \mathbf{n}_i \right).$$

The optimal weighting factor w_i in (18) is the inverse of the variance of $\mathbf{l}_i^T \tilde{\mathbf{a}}$, which is $w_i = 1/(\tilde{\mathbf{a}}^T \Lambda_i \tilde{\mathbf{a}})$. Note that the weight w_i involves the unknown \mathbf{a} . To overcome this difficulty, we can approximate w_i by $1/\text{trace}(\Lambda_i)$ for the first iteration and by recomputing w_i with the previously estimated \mathbf{a} in the subsequent iterations. Usually, two or three iterations are enough.

Visible fixed point. Since the fixed point is visible, we have N observations: $\{\mathbf{a}_i | i = 1, \dots, N\}$. We can therefore estimate \mathbf{a} by minimizing $\sum_{i=1}^N \|\mathbf{a} - \mathbf{a}_i\|^2$, assuming that the image points are detected with the same accuracy. The solution is simply $\mathbf{a} = (\sum_{i=1}^N \mathbf{a}_i) / N$.

The above estimation does not make use of the fact that the fixed point is also the intersection of the N observed lines of the 1D object. Therefore, a better technique to estimate \mathbf{a} is to minimize the following function:

$$\begin{aligned} \mathcal{F} &= \sum_{i=1}^N \left[(\mathbf{a} - \mathbf{a}_i)^T \mathbf{V}_i^{-1} (\mathbf{a} - \mathbf{a}_i) + w_i \|\mathbf{l}_i^T \tilde{\mathbf{a}}\|^2 \right] \\ &= \sum_{i=1}^N \left[(\mathbf{a} - \mathbf{a}_i)^T \mathbf{V}_i^{-1} (\mathbf{a} - \mathbf{a}_i) + w_i \|\mathbf{n}_i^T \mathbf{a} + q_i\|^2 \right], \end{aligned} \quad (19)$$

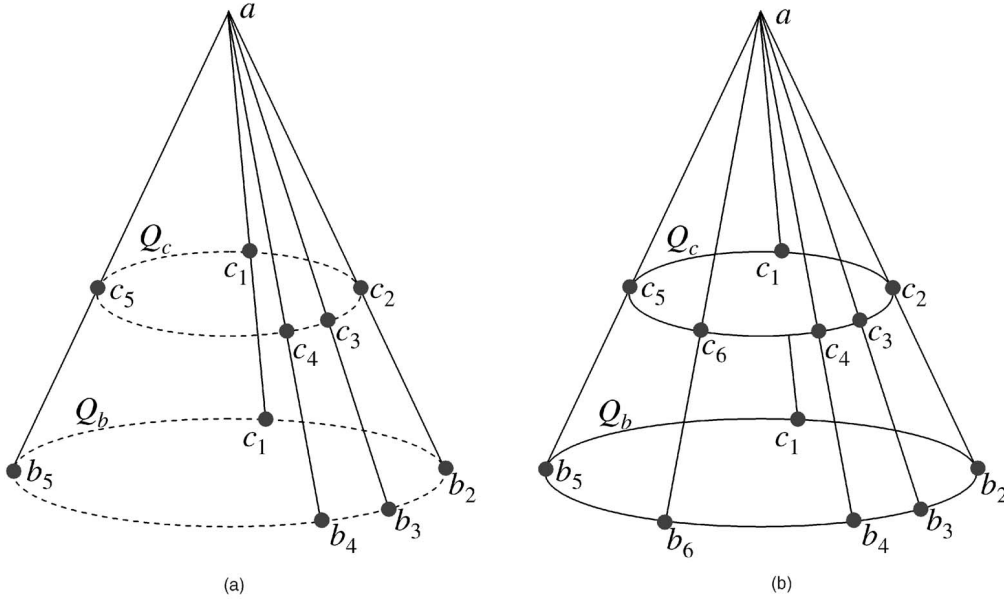


Fig. 2. Singularity of camera calibration with 1D objects.

where \mathbf{V}_i is the covariance matrix of the detected point \mathbf{a}_i . The derivative of the above function with respect to \mathbf{a} is given by

$$\frac{\partial \mathcal{F}}{\partial \mathbf{a}} = 2 \sum_{i=1}^N [\mathbf{V}_i^{-1}(\mathbf{a} - \mathbf{a}_i) + w_i \mathbf{n}_i \mathbf{n}_i^T \mathbf{a} + w_i q_i \mathbf{n}_i].$$

Setting it to 0 yields,

$$\mathbf{a} = \left(\sum_{i=1}^N (\mathbf{V}_i^{-1} + w_i \mathbf{n}_i \mathbf{n}_i^T) \right)^{-1} \left(\sum_{i=1}^N (\mathbf{V}_i^{-1} \mathbf{a}_i - w_i q_i \mathbf{n}_i) \right).$$

If more than three points are visible in each image, the known cross ratio provides an additional constraint in determining the fixed point.

For an accessible description of uncertainty manipulation, the reader is referred to [25, chapter 2].

3.5 Singularities

This calibration algorithm, like almost any algorithm, has singularities. It is important to be aware of the singularities in order to obtain reliable results, in practice, by avoiding them.

As we have said in Section 3.1, each observation of the 1D object provides one constraint on calibration and we need at least six independent observations in order to solve calibration. If the 1D calibration object moves in a particular configuration such that it depends on only five parameters, then it is a singularity for this calibration algorithm. Conversely, if any new observation depends on five observations, then the new observation does not provide additional constraint on calibration, the set of such observations form a singular configuration for calibration. Let's take a geometric construction approach.

Look at Fig. 2a. We have five observations: $\{(a, b_i, c_i) | i = 1, \dots, 5\}$. There exist a quadratic curve Q_b passing through $\{b_i | i = 1, \dots, 5\}$ and another quadratic curve Q_c passing through $\{c_i | i = 1, \dots, 5\}$. Now, look at Fig. 2b and consider a sixth observation (a, b_6, c_6) . If b_6 is on Q_b and c_6 is on Q_c , then

the sixth observation is not independent of the five first observations because c_6 is completely defined by the intersection between the line from a to b_6 and the quadratic curve Q_c . Therefore, this sixth observation does not provide any additional constraint on calibration. In other words, if all points b_i s are on the same quadratic curve Q_b and all points c_i s are on the same quadratic curve Q_c , then this is a singularity of calibration. Since the image of a quadratic curve in space is a quadratic curve, if we move the free end of the calibration object along a quadratic curve, we have a degenerate configuration for calibration. Since the free end can only move on a spherical surface, the quadratic curve is a circle. However, one can imagine that the calibration result will not be reliable if the free end moves on a nonplanar curve but close to be an ellipse. We should avoid this type of motions in practice.

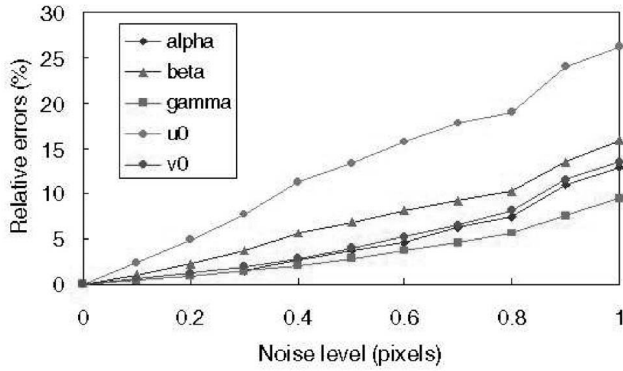
4 EXPERIMENTAL RESULTS

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

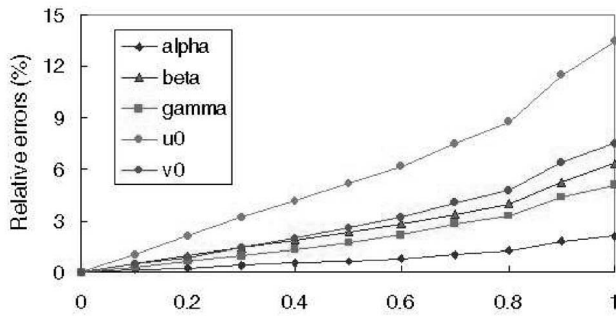
4.1 Computer Simulations

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 A at $[0, 35, 150]^T$. The other endpoint of the stick is B and C is located at the half way between A and 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 A, B, and 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 [19]. He pointed



(a)



(b)

Fig. 3. Calibration errors with respect to the noise level of the image points. (a) Closed-form solution. (b) Nonlinear optimization.

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. 3 are the average. Fig. 3a displays the relative errors of the closed-form solution while Fig. 3b 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 percent less errors). At 1 pixel noise level, the errors for the closed-form solution are about 12 percent, while those for the nonlinear minimization are about 6 percent.

We have also conducted an experiment with simulated data when the fixed point is outside of the camera's field of view. The setup is exactly the same as the previous experiment except the following changes: the fixed point A is at $[0, -50, 170]^T$; the stick length L is equal to 100; we have again generated 100 random orientations of the stick by sampling θ in $[\pi/6, 5\pi/6]$ and ϕ in $[\pi/6, 5\pi/6]$ according to uniform distribution. Note that we used a smaller range for ϕ than in the previous experiment, in order to keep the generated points visible to the camera. Noise with level ranging from 0.1 pixels to 1 pixel is added to the data. The projected fixed point a is around $(320, -54.12)$ and the fixed point is estimated as described in Section 3.4 for the invisible case. For each noise level, we perform 120 independent trials and the results

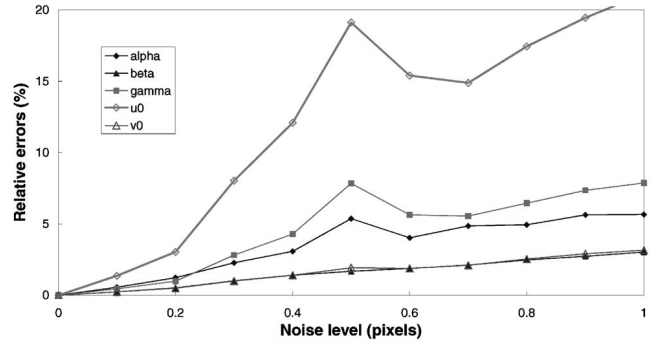


Fig. 4. Calibration errors with respect to the noise level of the image points when the fixed point is out of the camera's field of view.

shown in Fig. 4 are the average of the final nonlinear minimization results. Compared with Fig. 3b, the results with invisible fixed point are about twice as worse. We also noted the instability of the closed-form solution when the noise level is higher (about a quarter of trials failed for $\sigma = 1$). All these may be due to poorer estimation of the fixed point because of its implicit observation and to smaller coverage of working space because of smaller range for ϕ . This suggests that more than the minimum three points should be used in practice to achieve reasonable accuracy.

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 moved 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 are shown in Fig. 5. 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 degrees).

For comparison, we also used the plane-based calibration technique described in [23] to calibrate the same camera. Five images of a planar pattern were taken, and one of them is shown in Fig. 6. 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 percent.

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.

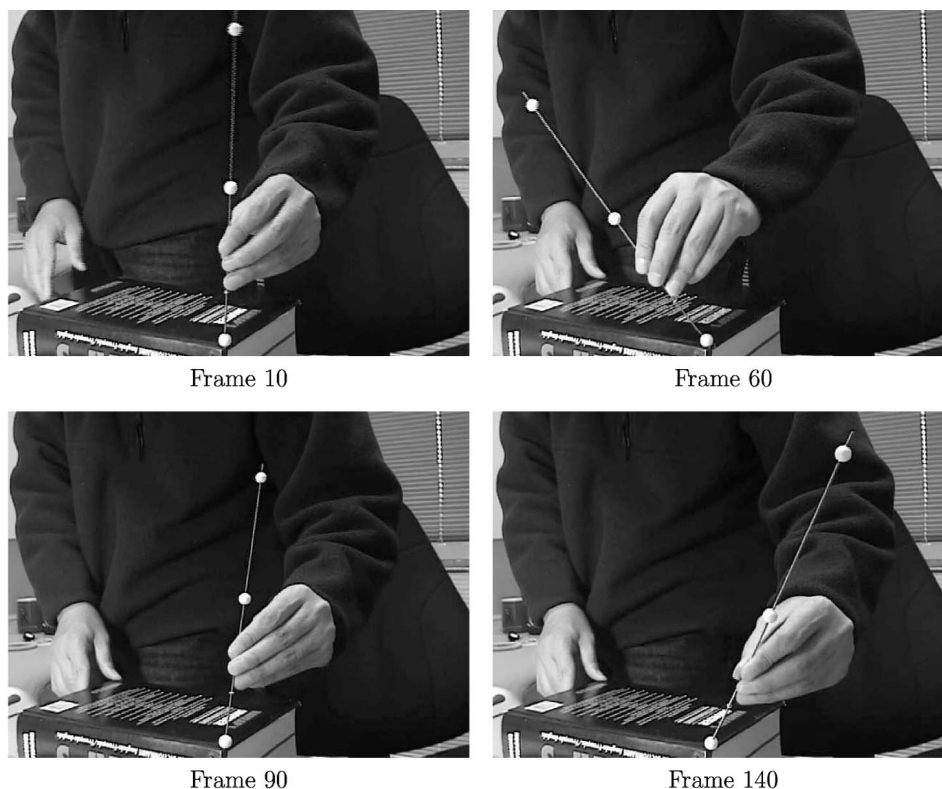


Fig. 5. Sample images of a 1D object used for camera calibration.

TABLE 1
Calibration Results with Real Data

<i>Solution</i>	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%

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

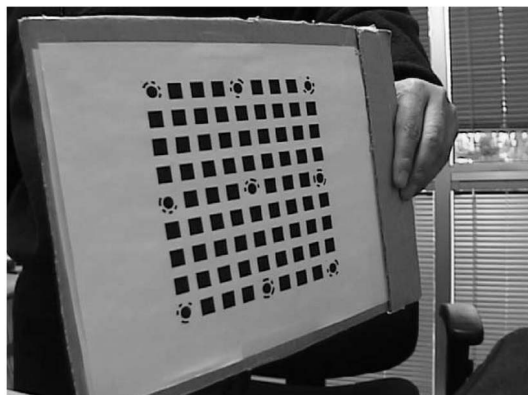


Fig. 6. A sample image of the planar pattern used for camera calibration.

cameras mounted apart from each other, where the calibration objects are required to be visible simultaneously.

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. This is a topic we should investigate in the future.

Although we have studied singularities of the proposed algorithm, a more thorough investigation may need to be pursued.

ACKNOWLEDGMENTS

The short version of this paper [24] was published in the *Proceedings of European Conference on Computer Vision*, 2002. The author would like to thank Peter Sturm for suggesting a solution to the case where the fixed point is not observed by the camera. The author would also like to thank the reviewers for suggesting conducting singularity analysis.

REFERENCES

- [1] D.C. Brown, "Close-Range Camera Calibration," *Photogrammetric Eng.*, vol. 37, no. 8, pp. 855-866, 1971.
- [2] B. Caprile and V. Torre, "Using Vanishing Points for Camera Calibration," *Int'l J. Computer Vision*, vol. 4, no. 2, pp. 127-140, Mar. 1990.
- [3] W. Faig, "Calibration of Close-Range Photogrammetry Systems: Mathematical Formulation," *Photogrammetric Eng. and Remote Sensing*, vol. 41, no. 12, pp. 1479-1486, 1975.
- [4] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
- [5] O. Faugeras, T. Luong, and S. Maybank, "Camera Self-Calibration: Theory and Experiments," *Proc. Second European Conf. Computer Vision*, G. Sandini, ed., vol. 588, pp. 321-334, May 1992.
- [6] O. Faugeras and G. Toscani, "The Calibration Problem for Stereo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 15-20, June 1986.
- [7] S. Ganapathy, "Decomposition of Transformation Matrices for Robot Vision," *Pattern Recognition Letters*, vol. 2, pp. 401-412, Dec. 1984.
- [8] D. Gennery, "Stereo-Camera Calibration," *Proc. 10th Image Understanding Workshop*, pp. 101-108, 1979.
- [9] R. Hartley, "Self-Calibration from Multiple Views with a Rotating Camera," *Proc. Third European Conf. Computer Vision*, J.-O. Eklundh, ed., vol. 800-801, pp. 471-478, May 1994.
- [10] R.I. Hartley, "An Algorithm for Self Calibration from Several Views," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 908-912, June 1994.
- [11] A. Heyden and M. Pollefeys, "Multiple View Geometry," *Emerging Topics in Computer Vision*, G. Medioni and S.B. Kang, eds., chapter 3, pp. 45-108, Prentice Hall, 2003.
- [12] D. Liebowitz and A. Zisserman, "Metric Rectification for Perspective Images of Planes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 482-488, June 1998.
- [13] Q.-T. Luong and O.D. Faugeras, "Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices," *Int'l J. Computer Vision*, vol. 22, no. 3, pp. 261-289, 1997.
- [14] Q.-T. Luong, "Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus Grande Autonomie des Systemes Robotiques," PhD thesis, Universite de Paris-Sud, Centre d'Orsay, Dec. 1992.
- [15] S.J. Maybank and O.D. Faugeras, "A Theory of Self-Calibration of a Moving Camera," *Int'l J. Computer Vision*, vol. 8, no. 2, pp. 123-152, Aug. 1992.
- [16] J.J. More, "The Levenberg-Marquardt Algorithm, Implementation and Theory," *Numerical Analysis*, G.A. Watson, ed., 1977.
- [17] G. Stein, "Accurate Internal Camera Calibration Using Rotation, with Analysis of Sources of Error," *Proc. Fifth Int'l Conf. Computer Vision*, pp. 230-236, June 1995.
- [18] P. Sturm and S. Maybank, "On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 432-437, June 1999.
- [19] B. Triggs, "Autocalibration from Planar Scenes," *Proc. Fifth European Conf. Computer Vision*, pp. 89-105, 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 J. Robotics and Automation*, vol. 3, no. 4, pp. 323-344, Aug. 1987.
- [21] G.Q. Wei and S.D. Ma, "A Complete Two-Plane Camera Calibration Method and Experimental Comparisons," *Proc. Fourth Int'l Conf. Computer Vision*, pp. 439-446, May 1993.
- [22] J. Weng, P. Cohen, and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965-980, Oct. 1992.
- [23] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.
- [24] Z. Zhang, "Camera Calibration with One-Dimensional Objects," *Proc. European Conf. Computer Vision*, vol. 4, pp. 161-174, May 2002.
- [25] Z. Zhang and O.D. Faugeras, *3D Dynamic Scene Analysis: A Stereo Based Approach*. Berlin, Heidelberg: Springer, 1992.



Zhengyou Zhang received the BS degree in electronic engineering from the University of Zhejiang, China, in 1985, the MS degree in computer science from the University of Nancy, France, in 1987, the PhD degree in computer science from the University of Paris XI, France, in 1990, and the Doctor of Science diploma (Habilitation à diriger des recherches) from the University of Paris XI, in 1994. The MS dissertation is on speech recognition, while the

PhD and Dsci Dissertations are on computer vision. He is a senior researcher at Microsoft Research, Redmond, Washington. He was with the French National Institute for Research in Computer Science and Control (INRIA) for 11 years until he joined Microsoft Research in March 1998. In 1996 to 1997, he spent a one year sabbatical as an invited researcher at the Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan. His current research interests include 3D computer vision, dynamic scene analysis, vision and graphics, facial image analysis, multisensory technology, and human-computer interaction. He is a senior member of the IEEE and is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *International Journal of Pattern Recognition and Artificial Intelligence*. He has published more than 100 papers in referred international journals and conferences and has coauthored the following books: *3D Dynamic Scene Analysis: A Stereo Based Approach* (Springer, Berlin, Heidelberg, 1992), *Epipolar Geometry in Stereo, Motion and Object Recognition* (Kluwer Academic, 1996), and *Computer Vision* (in Chinese, Chinese Academy of Sciences, 1998). He has been on the program committees for numerous international conferences and, currently, he is an area chair and a demo chair of the International Conference on Computer Vision (ICCV '03), October, 2003, Nice, France, and a program chair of the Asian Conference on Computer Vision (ACCV '04), January 2004, Jeju Island, Korea.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.