Geometric Camera Calibration Using Circular Control Points

Janne Heikkilä

Abstract—Modern CCD cameras are usually capable of a spatial accuracy greater than 1/50 of the pixel size. However, such accuracy is not easily attained due to various error sources that can affect the image formation process. Current calibration methods typically assume that the observations are unbiased, the only error is the zero-mean independent and identically distributed random noise in the observed image coordinates, and the camera model completely explains the mapping between the 3D coordinates and the image coordinates. In general, these conditions are not met, causing the calibration results to be less accurate than expected. In this paper, a calibration procedure for precise 3D computer vision applications is described. It introduces bias correction for circular control points and a nonrecursive method for reversing the distortion model. The accuracy analysis is presented and the error sources that can reduce the theoretical accuracy are discussed. The tests with synthetic images indicate improvements in the calibration results in limited error conditions. In real images, the suppression of external error sources becomes a prerequisite for successful calibration.

Index Terms—Camera model, lens distortion, reverse distortion model, calibration procedure, bias correction, calibration accuracy.

1 Introduction

In 3D machine vision, it necessary to know the relationship between the 3D object coordinates and the image coordinates. This transformation is determined in geometric camera calibration by solving the unknown parameters of the camera model. Initially, camera calibration techniques were developed in the field of photogrammetry for aerial imaging and surveying. First, photographic cameras were used, but recently video cameras have replaced them almost completely. Also new application areas, like robot vision and industrial metrology, have appeared, where camera calibration plays an important role.

Depending on the application, there are different requirements for camera calibration. In some applications, such as robot guidance, the calibration procedure should be fast and automatic, but in metrology applications, the precision is typically a more important factor. The traditional camera calibration procedures, such as bundle adjustment [1], are computationally greedy full-scale optimization approaches. Therefore, most of the calibration methods suggested during the last few decades in computer vision literature are mainly designed for speeding up the process by simplifying or linearizing the optimization problem. The well-known calibration method developed by Roger Tsai [2] belongs to this category. Other techniques also based on the linear transformation, for example [3], [4], [5], [6], are fast but quite inaccurate. The simplifications made reduce the precision of the parameter estimates and, as a consequence, they are not suitable for applications in 3D metrology as such.

• *The author is with the Machine Vision and Media Processing Unit, Infotech Oulu and Department of Electrical Engineering, FIN-90014 University of Oulu, Oulu, Finland. E-mail: jth@ee.oulu.fi.*

Manuscript received 4 Dec. 1998; revised 4 Oct. 1999; accepted 31 May 2000. Recommended for acceptance by K. Bowyer.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108389.

Due to the increased processing power of standard workstations, the nonlinear nature of the estimation problem is not as restricting as it was a few years ago. The calibration procedure can be accomplished in a couple of seconds iteratively. This gives us a good reason for improving the accuracy of the calibration methods without introducing a lot of extra time for computation. An accuracy of 1/50 of the pixel size (around 1/50,000 of the image size) is a realistic goal that can be achieved in low noise conditions with proper subpixel feature extraction techniques. The main improvements in the new calibration procedure presented in the following sections are the camera model, which allows accurate mapping in both directions, and the elimination of the bias in the coordinates of the circular control points.

In Section 2, we begin by describing the camera model for projection and back-projection. In Section 3, the projective geometry of the circular control points is reviewed and the necessary equations for mapping the circles into the image plane are presented. Section 4 describes a three-step calibration procedure for circular control points. Experimental results with the calibration procedure are reported in Section 5 and the effects of some typical error sources are discussed in Section 6. Finally, Section 7 offers concluding remarks.

2 Camera Model

In camera calibration, the transformation from 3D world coordinates to 2D image coordinates is determined by solving the unknown parameters of the camera model. Depending on the accuracy requirements, the model is typically based on either orthographic or perspective projection. Orthographic transformation is the roughest approximation assuming the objects in 3D space to be orthogonally projected on the image plane. It is more suitable for vision applications where the requirements of

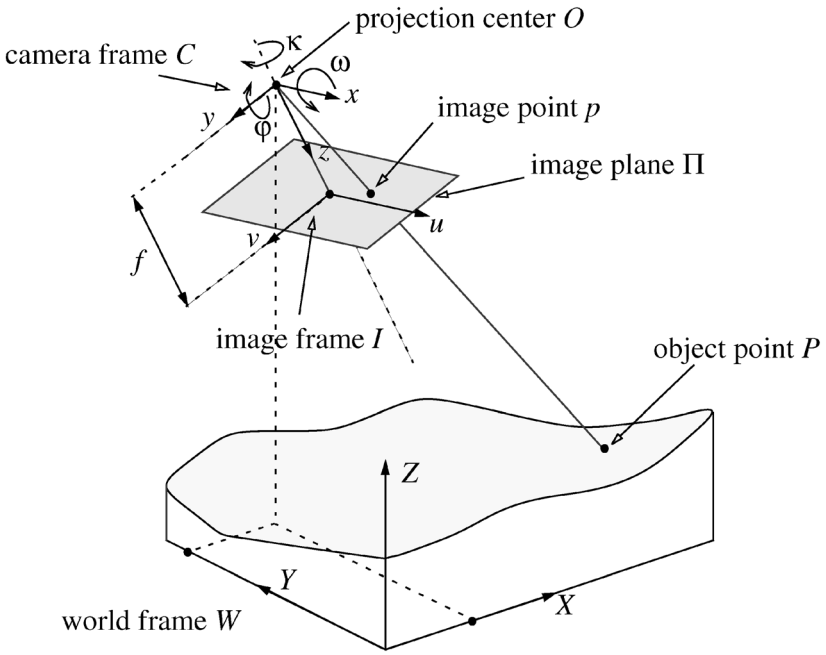


Fig. 1. Pinhole camera model.

the geometric accuracy are somewhat low. Due to linearity, it provides a simpler and computationally less expensive solution than perspective projection which is a nonlinear form of mapping. However, for 3D motion estimation and reconstruction problems, perspective projection gives an idealized mathematical framework, which is actually quite accurate for high quality camera systems. For off-the-shelf systems, the perspective projection model is often augmented with a lens distortion model.

Let us first consider a pure perspective projection (i.e., pinhole) camera model illustrated in Fig. 1. The center of projection is at the origin O of the camera frame C. The image plane Π is parallel to the xy plane and it is displaced with a distance f (focal length) from O along the z axis. The z axis is also called the optical axis, or the principal axis, and the intersection of Π and the optical axis is called the principal point o. The u and v axes of the 2D image coordinate frame I are parallel to the x and y axes, respectively. The coordinates of the principal point in I are $[u_0, v_0]^T$.

Let P be an arbitrary 3D point located on the positive side of the z axis and p its projection on Π. The coordinates of P in the camera frame C are $[x, y, z]^T$ and in the world frame W the coordinates are $[X, Y, Z]^T$. The coordinates of p in I are $[u, v]^T$ and they can be solved from the homogeneous coordinates given by the transformation

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \propto \begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \mathbf{F} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{PM} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{F}$ is the perspective transformation matrix (PTM),

$$\mathbf{P} = \begin{bmatrix} sf & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2}$$

λ is a scale factor, s is the aspect ratio, and $\mathbf{M}$ is a 4 by 4 matrix describing the mapping from W to C. It is decomposed as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3}$$

where $\mathbf{t} = [t_x, t_y, t_z]^T$ describes the translation between the two frames, and $\mathbf{R}$ is a 3 by 3 orthonormal rotation matrix which can be defined by the three Euler angles ω, φ, and κ. If $\mathbf{R}$ is known, these angles can be computed using, for example, the following decomposition [6]:

$$\begin{aligned} \varphi &= \sin^{-1} r_{31} \\ \omega &= \operatorname{atan}\, 2\left(-\frac{r_{32}}{\cos\varphi}, \frac{r_{33}}{\cos\varphi}\right) \\ \kappa &= \operatorname{atan}\, 2\left(-\frac{r_{21}}{\cos\varphi}, \frac{r_{11}}{\cos\varphi}\right), \end{aligned} \tag{4}$$

where r_{ij} is the entry from the *i*th row and the *j*th column of the matrix $\mathbf{R}$ and $\operatorname{atan2}(y, x)$ is the two-argument inverse tangent function giving the angle in the range $(-\pi, \pi]$. Because $\sin\varphi = \sin(\pi - \varphi)$, the Euler angles do not represent the rotation matrix uniquely. Hence, there are two equivalent decompositions for the matrix $\mathbf{R}$. As we can see, (4) has singularity if $r_{31} = \pm 1$, i.e., $\varphi = \pi/2$ or $\varphi = 3\pi/2$. In those cases, we can choose $\kappa = 0$, and $\omega = \operatorname{atan2}(r_{12}, r_{22})$, or vice versa [6]. In most situations, we could also prevent the singularities by carefully planning the calibration setup.

The parameters $t_x, t_y, t_z, \omega, \varphi,$ and κ are called *extrinsic parameters* or *exterior projective parameters* and the parameters s, f, u_0 and v_0 are the *intrinsic parameters* or *interior projective parameters* of the pinhole camera model.

It is usually more convenient to express the image coordinates in pixels. Therefore, the coordinates obtained from (1) are multiplied by factors D_u and D_v that specify the relationship between pixels and the physical object units,

for example, millimeters. However, knowing the precise values of these conversion factors is not necessary because they are linearly dependent on the parameters s and f that are adjusted during calibration.

In real cameras, perspective projection is typically not sufficient for modeling the mapping precisely. Ideally, the light rays coming from the scene should pass through the optical center linearly, but in practice, lens systems are composed of several optical elements introducing nonlinear distortion to the optical paths and the resulting images. The camera model of (1) produces the ideal image coordinates $[u, v]^T$ of the projected point p. In order to separate these errorless but unobservable coordinates from their observable distorted counterparts, we will henceforth denote the correct or corrected coordinates of (1) by $\mathbf{a}_c = [u_c, v_c]^T$ and the distorted coordinates by $\mathbf{a}_d = [u_d, v_d]^T$.

Several methods for correcting the lens distortion have been developed. The most commonly used approach is to decompose the distortion into radial and decentering components [7]. Knowing the distorted image coordinates $\mathbf{a}_d$, the corrected coordinates $\mathbf{a}_c$ are approximated by

$$\mathbf{a}_c = \mathbf{a}_d + \mathcal{F}_D(\mathbf{a}_d, \boldsymbol{\delta}), \tag{5}$$

where

$$\mathcal{F}_D(\mathbf{a}_d, \boldsymbol{\delta}) = \begin{bmatrix} \bar{u}_d(k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + \ldots) \\ \quad + \big(2p_1 \bar{u}_d \bar{v}_d + p_2(r_d^2 + 2\bar{u}_d^2)\big)(1 + p_3 r_d^2 + \cdots) \\ \bar{v}_d(k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + \ldots) \\ \quad + (p_1(r_d^2 + 2\bar{v}_d^2) + 2p_2 \bar{u}_d \bar{v}_d)(1 + p_3 r_d^2 + \cdots) \end{bmatrix}, \tag{6}$$

$$\bar{u}_d = u_d - u_0, \bar{v}_d = v_d - v_0, r_d = \sqrt{\bar{u}_d^2 + \bar{v}_d^2},$$

and $\boldsymbol{\delta} = [k_1, k_2, \ldots, p_1, p_2, \ldots]^T$. The parameters $k_1, k_2, \ldots$ are the coefficients for the radial distortion that causes the actual image point to be displaced radially in the image plane, and the parameters $p_1, p_2, \ldots$ are the coefficients for the decentering or tangential distortion which may occur when the centers of the curvature of the lens surfaces in the lens system are not strictly collinear.

Other distortion models have also been proposed in the literature. For example, Melen [6] used a model where $\mathbf{P}$ in (2) was augmented with terms for linear distortion. This is useful if the image axes are not orthogonal, but in most cases the CCD arrays are almost perfect rectangles, and hence, the linear distortion component is insignificant (less than 0.01 pixels [8]). In the calibration method proposed by Faugeras and Toscani [5], the geometric distortion was corrected using bilinear transformation in small image regions.

For camera calibration, it is favorable to find a transformation from the 3D world coordinates to the real image coordinates. This enables us to use a least-squares technique as an optimal estimator of the camera parameters. Directly applying (1) and (5) implies that we first need to correct the distortion and then estimate the camera parameters of (1). An obvious problem is that the distortion coefficients are not usually known in advance and due to strong coupling, they cannot be reliably estimated without knowing the other camera parameters.

In the literature, there are several solutions to overcome this problem. Tsai [2] and Lenz and Tsai [9] decomposed the camera model into linear and nonlinear parts where the parameters are decoupled. However, only radial distortion can be used and the solution is not optimal. Weng et al. [10] suggested an iterative scheme where the parameters of the distortion model and the projection model are fixed, in turn, and estimated separately. A commonly used approach in photogrammetry [7] is to perform a full-scale optimization for all parameters by minimizing the sum of squared errors between the corrected image coordinates and the synthetic coordinates given by the camera model. In practice, this means that (6) is evaluated with noisy observations, which may deteriorate the calibration result. In order to minimize the error between the observed and model coordinates, the distorted image coordinates should be expressed in terms of their undistorted counterparts. For this, we need an inverse distortion model. As can be easily noticed, there is no analytic solution for the inverse problem, and thus, we need to approximate it. Melen [6], for example, used the following model:

$$\mathbf{a}_d \approx \mathbf{a}_c - \mathcal{F}_D(\mathbf{a}_c, \boldsymbol{\delta}). \tag{7}$$

The fitting results given by this model are often satisfactory, because the distortion coefficients are typically small values causing the model to be almost linear. It should be noticed that the optimal distortion coefficients in a least squares sense are different for (5) and (7).

Another solution is to create the following recursion based on (5):

$$\begin{aligned} \mathbf{a}_d &\approx \mathbf{a}_c - \mathcal{F}_D(\mathbf{a}_d, \boldsymbol{\delta}) \approx \mathbf{a}_c - \mathcal{F}_D(\mathbf{a}_c - \mathcal{F}_D(\mathbf{a}_d, \boldsymbol{\delta}), \boldsymbol{\delta}) \\ &\approx \mathbf{a}_c - \mathcal{F}_D(\mathbf{a}_c - \mathcal{F}_D(\mathbf{a}_c - \mathcal{F}_D(\mathbf{a}_d, \boldsymbol{\delta}), \boldsymbol{\delta}), \boldsymbol{\delta}) \approx \cdots. \end{aligned} \tag{8}$$

The error introduced when substituting $\mathbf{a}_d$ with $\mathbf{a}_c$ on the right-hand side gets smaller for each iteration. In practice, at least three or four iterations are required to compensate for strong lens distortions. This means that the distortion function $\mathcal{F}_D$ is evaluated several times in different locations of the image, which makes this technique less attractive.

In order to avoid extensive computation, we can take a first order Taylor series approximation of $\mathcal{F}_D$ about $\mathbf{a}_c$:

$$\mathbf{a}_c \approx \mathbf{a}_d + \mathcal{F}_D(\mathbf{a}_c, \boldsymbol{\delta}) + \mathbf{D}(\mathbf{a}_c)(\mathbf{a}_d - \mathbf{a}_c), \tag{9}$$

where

$$\mathbf{D}(\mathbf{a}_c) = \left[\frac{\partial}{\partial u}\mathcal{F}_D(\mathbf{a}, \boldsymbol{\delta}) \frac{\partial}{\partial v}\mathcal{F}_D(\mathbf{a}, \boldsymbol{\delta})\right]\Big|_{\mathbf{a}=\mathbf{a}_c} \tag{10}$$

Solving $\mathbf{a}_d$ from (9) yields

$$\mathbf{a}_d \approx \mathbf{a}_c - (\mathbf{I} + \mathbf{D}(\mathbf{a}_c))^{-1} \mathcal{F}_D(\mathbf{a}_c, \boldsymbol{\delta}). \tag{11}$$

The elements of $\mathbf{D}(\mathbf{a}_c)$ are small ($<< 1$) which makes it possible to use the following approximation:

$$\mathbf{a}_d \approx \mathbf{a}_c - \frac{1}{d_{11}(\mathbf{a}_c) + d_{22}(\mathbf{a}_c) + 1} \mathcal{F}_D(\mathbf{a}_c, \boldsymbol{\delta}) \equiv \mathbf{a}_c - \mathcal{F}_D^*(\mathbf{a}_c, \boldsymbol{\delta}), \tag{12}$$

where $d_{11}(\mathbf{a}_c)$ and $d_{22}(\mathbf{a}_c)$ are the upper left and lower right elements of $\mathbf{D}(\mathbf{a}_c)$, respectively. If only two radial distortion coefficients k_1 and k_2, and two decentering distortion coefficients p_1 and p_2 are used, the approximation of the inverse distortion model becomes

$$\mathcal{F}_D^*(\mathbf{a}_c, \boldsymbol{\delta}) = \frac{1}{4k_1 r_c^2 + 6k_2 r_c^4 + 8p_1\bar{v}_c + 8p_2\bar{u}_c + 1}\mathcal{F}_D(\mathbf{a}_c, \boldsymbol{\delta}), \tag{13}$$

where $r_c = \sqrt{\bar{u}_c^2 + \bar{v}_c^2}$. If necessary, extending this model with higher order terms is straightforward.

By replacing $u = u_c$ and $v = v_c$ in (1) and combining it with (12), we obtain a *forward* camera model which converts the 3D world coordinates to distorted image coordinates.

Using a *backward* camera model, we can transform the distorted camera coordinates to lines of sight in the 3D world coordinate system, or to the intersections of these lines with a known 2D plane. Let us assume that we have a 2D plane Π' with a coordinate system H, whose origin is at $\mathbf{h}_0 = [X_0, Y_0, Z_0]^T$, and it is spanned by the 3D vectors $\mathbf{h}_1 = [X_1, Y_1, Z_1]^T$ and $\mathbf{h}_2 = [X_2, Y_2, Z_2]^T$. The transformation from the corrected image coordinates $\mathbf{a}_c = [u_c, v_c]^T$ produced by (5) on the plane Π' can be expressed as

$$\lambda\begin{bmatrix}\mathbf{x}_H \\ 1\end{bmatrix} = (\mathbf{FH})^{-1}\begin{bmatrix}\mathbf{a}_c \\ 1\end{bmatrix}, \tag{14}$$

where λ is a scale factor, $\mathbf{x}_H = [X_H, Y_H]^T$ are the back projected coordinates in H and

$$\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_0 \\ 0 & 0 & 1\end{bmatrix}.$$

Due to the approximations made, $\mathcal{F}_D^*$ is not the exact inverse function of $\mathcal{F}_D$. Therefore, it may be necessary to use slightly different parameters $\boldsymbol{\delta}'$ in (6) than in (13). A method for estimating these parameters will be presented in Section 4.

3 Circular Control Points

Lines in the object space are mapped as lines on the image plane, but in general perspective projection is not a shape preserving transformation. Two- and three-dimensional shapes are deformed if they are not coplanar with the image plane. This is also true for circular landmarks, which are commonly used control points in calibration. However, a bias between the observations and the model is induced if the centers of their projections in the image are treated as projections of the circle centers. In this section, we will review the necessary equations for using the centers of the projections as image observations without introducing any bias.

Let us assume that a circular control point R with radius r is located on the plane Π' so that its center is at the origin of the planar coordinate frame H. Circles are quadratic curves that can be expressed in the following manner:

$$AX_H^2 + 2BX_HY_H + CY_H^2 + 2DX_H + 2EY_H + F = 0, \tag{15}$$

where A, B, C, D, E, and F are coefficients that define the shape and location of the curve. In homogeneous coordinates, this curve can be written as

$$\begin{bmatrix}\mathbf{x}_H \\ 1\end{bmatrix}^{\mathrm{T}}\mathbf{Q}\begin{bmatrix}\mathbf{x}_H \\ 1\end{bmatrix} = 0, \tag{16}$$

where

$$\mathbf{Q} = \begin{bmatrix}A & B & D \\ B & C & E \\ D & E & F\end{bmatrix}.$$

For the circle R,

$$\mathbf{Q} = \begin{bmatrix}-1/r^2 & 0 & 0 \\ 0 & -1/r^2 & 0 \\ 0 & 0 & 1\end{bmatrix}.$$

Using (14), R can be projected on the image plane. The resulting curve becomes

$$\begin{bmatrix}\mathbf{a}_c \\ 1\end{bmatrix}^{\mathrm{T}}\mathbf{S}\begin{bmatrix}\mathbf{a}_c \\ 1\end{bmatrix} = 0, \tag{17}$$

where

$$\mathbf{S} = ((\mathbf{FH})^{-1})^{\mathrm{T}}\mathbf{Q}(\mathbf{FH})^{-1}. \tag{18}$$

We can see that the result is a quadratic curve, whose geometric interpretation is a circle, hyperbola, parabola, or ellipse. In practice, due to the finite rectangle corresponding to the image plane, the projection will be an ellipse, or in some special cases, a circle. As shown in [11], the center of the ellipse $\mathbf{e}_c = [u_e, v_e]^T$ can be obtained from

$$\lambda\begin{bmatrix}\mathbf{e}_c \\ 1\end{bmatrix} = \mathbf{S}^{-1}\begin{bmatrix}0 \\ 0 \\ 1\end{bmatrix}. \tag{19}$$

By combining (18) and (19), we get

$$\lambda\begin{bmatrix}\mathbf{e}_c \\ 1\end{bmatrix} = \mathbf{FGf}_3, \tag{20}$$

where $\mathbf{G} = \mathbf{HQ}^{-1}\mathbf{H}^T$, and $\mathbf{f}_3$ is a vector consisting of the last row of $\mathbf{F}$. The matrix $\mathbf{G}$ specifies the geometry, position, and orientation of the control point, and the perspective transformation matrix $\mathbf{F}$ specifies the mapping from the world coordinates to undistorted image coordinates that can be further converted to distorted image coordinates $\mathbf{e}_d$ using (12). As a result, we get

$$\mathbf{e}_d = \mathbf{e}_c - \mathcal{F}_D^*(\mathbf{e}_c, \boldsymbol{\delta}). \tag{21}$$

It should be noticed that using (20) instead of (1), we obtain an unbiased relationship between the observed centers of the ellipses and the camera model.

4 Calibration Procedure

The calibration procedure presented next is mainly intended to be used with circular landmarks. However, it is also suitable for small points without any specific geometry. In that case, the radius is set to zero. It is assumed that the camera model includes eight intrinsic

parameters $\boldsymbol{\theta}_{int} = [s, f, u_0, v_0, k_1, k_2, p_1, p_2]^{\mathrm{T}}$ and six extrinsic parameters $\boldsymbol{\theta}_{ext} = [t_x, t_y, t_z, \omega, \varphi, \kappa]^{\mathrm{T}}$. If necessary, adding higher order terms to the model is straightforward. In multiframe calibration, each frame has separate extrinsic parameters, but common intrinsic parameters. The radius of control points r and the conversion factors D_u and D_v are assumed to be known in advance, as well as the 3D coordinates, and the observed image coordinates of the control points.

Due to the nonlinear mapping of the camera model, there are no direct estimators of all the camera parameters that would produce an optimal solution in a least squares sense. We therefore use an iterative searching technique to estimate the vector

$$\boldsymbol{\theta} = \left[\boldsymbol{\theta}_{int}^{\mathrm{T}}, \boldsymbol{\theta}_{ext}^{\mathrm{T}}(1), \boldsymbol{\theta}_{ext}^{\mathrm{T}}(2), \cdots, \boldsymbol{\theta}_{ext}^{\mathrm{T}}(K)\right]^{\mathrm{T}}, \tag{22}$$

where K is the number of the frames. Before using the estimator, we need to have initial values of the camera parameters for guaranteeing that a global minimum can be achieved. This problem is solved by ignoring the lens distortion and using a linear estimator for the rest of the parameters.

Step 1: Initialization. Many variations of the linear calibration technique have been proposed in the literature, for example, [3], [4], [5], [6]. However, they are all based on the same idea where the perspective transformation matrix **F** is first estimated and then decomposed into intrinsic and extrinsic camera parameters. These methods provide a closed-form solution to the camera parameters. The major shortcoming is that they are not optimal estimators, because they do not produce minimum variance estimates. Another problem is that they ignore the effects of the radial and decentering lens distortion. Despite these shortcomings, linear calibration techniques can provide a good starting point for iterative search.

Several well-known procedures for estimating **F** are available, e.g., [5]. The main difference between the linear calibration methods is that how **F** is decomposed into camera parameters. For example, Faugeras and Toscani [5] extracted five intrinsic and six extrinsic parameters using a set of linear equations and Melen [6] used QR decomposition for estimating six intrinsic and six extrinsic parameters. In principle, these techniques can be directly applied to obtain the initial estimate for the iterative optimization step. However, due to the shortcomings listed above, these estimators tend to be rather sensitive to observation noise. Even small inaccuracy in **F** can cause significant errors especially in the intrinsic parameters. As a consequence, optimization may fail.

In the initialization step, where the aim is to produce the first, not the final, estimates of the camera parameters, it is often more reliable to use the nominal values for the focal length, the aspect ratio and the image center as the intrinsic parameters. With these values, we can directly write the projection matrix **P** using (2). Let us assume that the control points are not coplanar, which means that we have a full 3 by 4 estimate of the perspective transformation matrix **F** denoted by $\hat{\mathbf{F}}$. Thus, we can write the following equation:

$$\hat{\mathbf{F}} = \mathbf{P}\hat{\mathbf{M}} = \left[\mathbf{P}_{13}\hat{\mathbf{R}} \quad \mathbf{P}_{13}\hat{\mathbf{t}}\right], \tag{23}$$

where $\mathbf{P}_{13}$ is a matrix containing the first three columns of **P**, $\hat{\mathbf{R}}$, and $\hat{\mathbf{t}}$ are the estimates of **R** and **t**, respectively. Now, it is straightforward to get

$$\hat{\mathbf{R}} = \mathbf{P}_{13}^{-1}\hat{\mathbf{F}}_{13} \tag{24}$$

and

$$\hat{\mathbf{t}} = \mathbf{P}_{13}^{-1}\hat{\mathbf{F}}_{4}, \tag{25}$$

where $\hat{\mathbf{F}}_{13}$ is a matrix containing the first three columns, and $\hat{\mathbf{F}}_4$ the last column of $\hat{\mathbf{F}}$.

The matrix $\hat{\mathbf{R}}$ does not satisfy the orthonormality constraint of a standard rotation matrix, but we can normalize and orthogonalize it using the singular value decomposition (SVD):

$$\hat{\mathbf{R}} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}. \tag{26}$$

The orthonormal version of $\hat{\mathbf{R}}$ is given by

$$\hat{\mathbf{R}}' = \mathbf{U}\Sigma'\mathbf{V}^{\mathrm{T}}, \tag{27}$$

where Σ' is a diagonal matrix with diagonal elements 1, 1, and $|\mathbf{U}\mathbf{V}^{\mathrm{T}}|$ in descending order [11]. The Euler angles can now be extracted from $\hat{\mathbf{R}}'$ using (4).

If all the control points are coplanar, only a submatrix of **F** can be determined. Basically, the procedure is the same as above, with the exception that only two columns of $\hat{\mathbf{R}}'$ are estimated. The third column is easily obtained by utilizing the orthogonality of the basis vectors. If multiframe calibration is used, the initialization step is performed separately for every frame. The initial parameters are collected into a vector

$$\hat{\boldsymbol{\theta}}_0 = \left[1, f_0, N_u/2, N_v/2, 0, 0, 0, 0, \boldsymbol{\theta}_{ext}^{\mathrm{T}}(1), \boldsymbol{\theta}_{ext}^{\mathrm{T}}(2), \cdots, \boldsymbol{\theta}_{ext}^{\mathrm{T}}(K)\right]^{\mathrm{T}}, \tag{28}$$

where f_0 is the nominal focal length, N_u and N_v are the horizontal and vertical dimensions of the image.

Step 2: Iterative search. In this step, the parameters of the forward camera model are estimated by minimizing the weighted sum of squared differences between the observations and the model. Let us assume, that there are N circular control points and K images that are indexed by $n = 1, \ldots, N$ and $k = 1, \ldots, K$. A vector containing the observed image coordinates of the center of the ellipse n in the frame k is denoted by $\mathbf{e}_o(n, k)$, and the corresponding vector produced by the forward camera model of (20) and (21) is denoted by $\mathbf{e}_d(n, k)$. Now, the objective function can be expressed as

$$J(\boldsymbol{\theta}) = \mathbf{y}^{\mathrm{T}}(\boldsymbol{\theta})\mathbf{C}_{\mathrm{e}}{}^{-1}\mathbf{y}(\boldsymbol{\theta}), \tag{29}$$

where

$$\mathbf{y}(\boldsymbol{\theta}) = \Big[\big(\mathbf{e}_o(1,1)-\mathbf{e}_d(1,1)\big)^{\mathrm{T}}, \big(\mathbf{e}_o(2,1)-\mathbf{e}_d(2,1)\big)^{\mathrm{T}}, \cdots, \big(\mathbf{e}_o(N,K)-\mathbf{e}_d(N,K)\big)^{\mathrm{T}}\Big]^{\mathrm{T}}$$

and $\mathbf{C_e}$ is the covariance matrix of the observation error. There are various sources that can affect $\mathbf{C_e}$. Some of them are discussed later, in Section 6. If the covariance matrix is unknown or the measurement noise for individual observations is statistically independent and identically distributed, $\mathbf{C_e}^{-1}$ in (29) can be omitted. In general, the assumption of the statistically independent and identically distributed noise is not fulfilled, because the error variance depends on the size of the ellipse in the image, and the covariance between the horizontal and vertical coordinates is nonzero if the ellipse is inclined. As a consequence, the measurement noise covariance matrix $\mathbf{C_e}$ can be expressed as

$$\mathbf{C_e} = \begin{bmatrix} \mathbf{C_e}(1,1) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_e}(2,1) & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C_e}(N,K) \end{bmatrix}, \tag{30}$$

where

$$\mathbf{C_e}(n,k) = E\Big\{\big[\mathbf{e}_o(n,k)-E\{\mathbf{e}_o(n,k)\}\big]\big[\mathbf{e}_o(n,k)-E\{\mathbf{e}_o(n,k)\}\big]^{\mathrm{T}}\Big\}.$$

If $\mathbf{C_e}(n, k)$ is not known, it can be estimated. In practice, this means that we need to take several images from the same position and orientation, and then calculate the sample covariance in the following manner:

$$\hat{\mathbf{C}}_{\mathbf{e}}(n,k) = \frac{1}{W-1}\sum_{i=1}^{W}\Big[\mathbf{e}_o(n,k,i)-\overline{\mathbf{e}_o(n,k)}\Big]\Big[\mathbf{e}_o(n,k,i)-\overline{\mathbf{e}_o(n,k)}\Big]^{\mathrm{T}}, \tag{31}$$

where W is the number of the images used for estimating $\mathbf{C_e}(n, k)$, $\mathbf{e}_o(n, k, i)$ is the observed image coordinates (ellipse n, pose k, frame i), and $\overline{\mathbf{e}_o(n,k)}$ is the vector of the average coordinates. An obvious problem of this approach is that a large number of images needs to be taken for single calibration.

The parameters of the forward camera model are obtained by minimizing $J(\boldsymbol{\theta})$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta} J(\boldsymbol{\theta}). \tag{32}$$

There are several numerical techniques, such as the Levenberg-Marquardt method [12], that can be applied for the optimization problem of (32). With an efficient technique, only a few iterations are typically needed to attain the minimum.

Step 3: Backward camera model. In many applications, the observed image coordinates need to be projected back to 3D coordinates. This procedure was already described in Section 2. However, the reverse distortion model of (13), used in Step 2, is not exactly the inverse of (6). In order to get more consistent results in both directions, we estimate the parameter vector of (6) separately. The other intrinsic parameters are the same for both forward and backward models.

If the Steps 1 and 2 have been completed, the parameters of the forward model are already available. Using this model, we can produce a set of distorted points $\{\mathbf{a}_d(i)\}$ for arbitrary points $\{\mathbf{a}_c(i)\}$, where $i = 1, \ldots, M$. The points $\{\mathbf{a}_c(i)\}$ are generated so that they form an equally spaced 2D grid of about 1,000 to 2,000 points, for example, 40 by 40. Points must cover the entire image area and the outermost points should be approximately 5-10 percent outside the effective image region in order to get accurate mapping also close to the image borders. If the inversion were exact, these points would fulfill (5) expressed in the following matrix form:

$$\begin{bmatrix} \mathbf{a}_c(1)-\mathbf{a}_d(1) \\ \mathbf{a}_c(2)-\mathbf{a}_d(2) \\ \vdots \\ \mathbf{a}_c(M)-\mathbf{a}_d(M) \end{bmatrix} = \begin{bmatrix} \mathbf{B}(1) \\ \mathbf{B}(2) \\ \vdots \\ \mathbf{B}(M) \end{bmatrix}\boldsymbol{\delta}, \tag{33}$$

where

$$\mathbf{B}(i) = \begin{bmatrix} \bar{u}_d(i)r_d^2(i) & \bar{u}_d(i)r_d^4(i) & \ldots & 2\bar{u}_d(i)\bar{v}_d(i) & r_d^2(i)+2\bar{u}_d^2(i) & \ldots \\ \bar{v}_d(i)r_d^2(i) & \bar{v}_d(i)r_d^4(i) & \ldots & r_d^2(i)+2\bar{v}_d^2(i) & 2\bar{u}_d(i)\bar{v}_d(i) & \ldots \end{bmatrix}$$

$$\bar{u}_d(i) = u_d(i) - u_0, \bar{v}_d(i) = v_d(i) - v_0,$$

and

$$r_d(i) = \sqrt{\bar{u}_d^2(i) + \bar{v}_d^2(i)}.$$

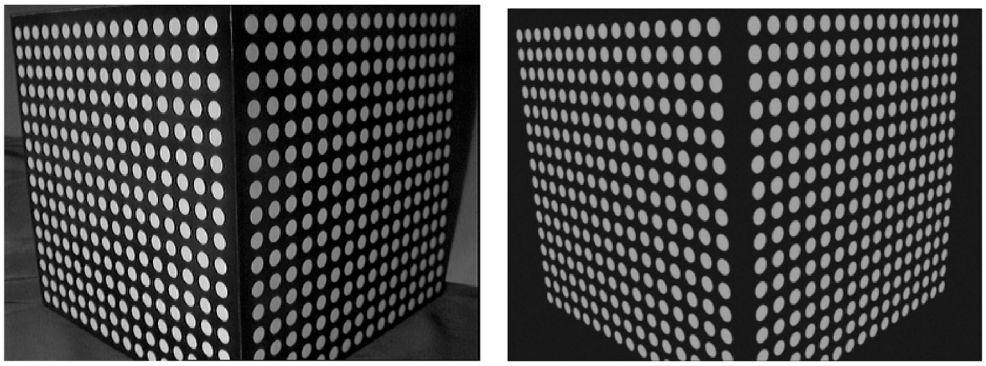
Due to the approximations made in derivation of the inverse distortion model, (33) does not hold exactly. Therefore, we need to adjust the parameter vector $\boldsymbol{\delta}$ for back projection. This can be performed using the following least-squares formulation:

$$\hat{\boldsymbol{\delta}}' = \begin{bmatrix} \mathbf{B}(1) \\ \mathbf{B}(2) \\ \vdots \\ \mathbf{B}(M) \end{bmatrix}^{+}\begin{bmatrix} \mathbf{a}_c(1)-\mathbf{a}_d(1) \\ \mathbf{a}_c(2)-\mathbf{a}_d(2) \\ \vdots \\ \mathbf{a}_c(M)-\mathbf{a}_d(M) \end{bmatrix}, \tag{34}$$

where $\hat{\boldsymbol{\delta}}'$ is the vector of the distortion parameters to be used in the backward camera model, and $[\ .\ ]^{+}$ denotes the pseudoinverse of the matrix [13].

5 Experiments

The camera calibration procedure suggested in Section 4 was tested in two experiments. First, the parameter estimates of the forward camera model were analyzed statistically using synthetic images and the results were compared with the outcome of a corresponding real image. Second, the precision of the inverse distortion model of (13) was evaluated by correcting and distorting random image coordinates.



(a) (b)

Fig. 2. Calibration images: (a) real and (b) synthetic.

5.1 Calibrating the Forward Camera Model

The tests were performed with 200 synthetic images and one real image that was used as a reference model. This real image shown in Fig. 2a was captured using an off-the-shelf monochrome CCD camera (Sony SSC-M370CE) equipped with a 6.3 mm by 4.7 mm image sensor and an 8.5 mm Cosmicar TV lens. The image was digitized from an analog CCIR type video signal using a Sunvideo frame grabber. The size of the image is 768 by 576 pixels, and the maximum dynamics is 256 gray levels. In the image, there is a

TABLE 1
Estimated Bias with Confidence Limits

parameter unit	true value	estimated bias conf. min. (99%) conf. max. (99%)		
		method A	method B	method C
s_u	1.00377723	-0.00030584 -0.00030696 -0.00030473	0.00003237 0.00003126 0.00003348	0.00003740 0.00003648 0.00003832
f [mm]	8.345900	0.005178 0.005122 0.005233	-0.000064 -0.000121 -0.000007	-0.000168 -0.000213 -0.000124
u_0 [pixels]	367.3353	-0.0096 -0.0170 -0.0022	0.0090 0.0017 0.0163	0.0221 0.0162 0.0280
v_0 [pixels]	305.9960	0.4879 0.4775 0.4983	-0.0175 -0.0280 -0.0070	-0.0343 -0.0429 -0.0256
k_1 ($\cdot 10^{-3}$) [mm^{-2}]	3.29545	-0.06682 -0.06730 -0.06634	-0.00953 -0.01001 -0.00904	-0.01034 -0.01079 -0.00988
k_2 ($\cdot 10^{-5}$) [mm^{-4}]	-2.6287	0.4829 0.4796 0.4863	0.0902 0.0870 0.0934	0.1009 0.0977 0.1040
p_1 ($\cdot 10^{-5}$) [mm^{-1}]	-0.13	-1.996 -2.030 -1.962	-1.386 -1.419 -1.352	-1.370 -1.401 -1.339
p_2 ($\cdot 10^{-5}$) [mm^{-1}]	4.065	-1.228 -1.249 -1.206	-1.201 -1.222 -1.180	-1.237 -1.257 -1.217

TABLE 2
Estimated Standard Deviation with Confidence Limits

parameter unit	theoretical std	estimated std conf. min. (95%) conf. max. (95%)		
		method A	method B	method C
s_u	0.00000417	0.00000802 0.00000731 0.00000890	0.00000797 0.00000726 0.00000885	0.00000659 0.00000601 0.00000732
f [mm]	0.000210	0.000398 0.000362 0.000442	0.000410 0.000374 0.000456	0.000321 0.000292 0.000356
u_0 [pixels]	0.0261	0.0530 0.0483 0.0588	0.0522 0.0476 0.0580	0.0424 0.0386 0.0471
v_0 [pixels]	0.0366	0.0748 0.0682 0.0831	0.0753 0.0687 0.0836	0.0622 0.0567 0.0691
k_1 ($\cdot 10^{-3}$) [mm^{-2}]	0.00196	0.00347 0.00316 0.00385	0.00347 0.00317 0.00386	0.00328 0.00299 0.00365
k_2 ($\cdot 10^{-5}$) [mm^{-4}]	0.0138	0.0239 0.0218 0.0265	0.0233 0.0212 0.0259	0.0227 0.0207 0.0253
p_1 ($\cdot 10^{-5}$) [mm^{-1}]	0.132	0.243 0.221 0.270	0.242 0.220 0.268	0.222 0.203 0.247
p_2 ($\cdot 10^{-5}$) [mm^{-1}]	0.087	0.154 0.141 0.171	0.148 0.135 0.165	0.144 0.131 0.160

calibration object which has two perpendicular planes, each with 256 circular control points. The centers of these points were located using the moment and curvature preserving ellipse detection technique [14] and renormalization conic fitting [15]. The calibration procedure was first applied to estimate the camera parameters based on the real image. The synthetic images were then produced using ray tracing with the camera parameters obtained from calibration and the known 3D model of the control points. In order to make the synthetic images better correspond to the real images, their intensity values were perturbed with additive Gaussian noise ($\sigma = 2$), and blurred using a 3 by 3 Gaussian filter ($\sigma = 1$ pixel). A sample image is shown in Fig. 2b.

With the synthetic images, we restrict the error sources only to quantization and random image noise. Ignoring all the other error sources is slightly unrealistic, but in order to achieve extremely accurate calibration results, these error sources should be eliminated or at least minimized somehow. The advantage of using simulated images instead of real images is that we can estimate some statistical properties of the procedure.

The control points from the synthetic images were extracted using the same techniques as with the real image, i.e., subpixel edge detection and ellipse fitting. Three different calibration methods were applied separately for all point sets. The first method (method A) is the traditional camera calibration approach which does not assume any geometry for the control points. In the second method (method B), circular geometry is utilized and all the observations are equally weighted. In the third method (method C), each observation is weighted by the inverse of the observation error covariance matrix $\mathbf{C}_\mathbf{e}$ that was estimated using (31) and all 200 images. Table 1 shows the estimated bias (average error) of the intrinsic parameters for all three methods. Confidence intervals with a 95 percent confidence level are also presented. As we can notice, all three methods produced slightly biased estimates. However, by regarding the circular geometry of the control points we can reduce the bias significantly. The remaining error is caused by the subpixel ellipse detector which is not completely unbiased. However, the effect of

the remaining error is so small that it does not have any practical significance.

The estimated standard deviation of the intrinsic parameters are given in Table 2. The theoretical accuracy that can be attained is approximated using the following equation:

$$\mathbf{C}_{\boldsymbol{\theta}} = E\left\{(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\right\} \approx \left[\mathbf{L}^{\mathrm{T}}(\hat{\boldsymbol{\theta}})\mathbf{C}_{\mathbf{e}}{}^{-1}\mathbf{L}(\hat{\boldsymbol{\theta}})\right]^{-1}, \qquad (35)$$

where $\mathbf{L}(\hat{\boldsymbol{\theta}})$ is the Jacobian matrix of $\mathbf{y}(\boldsymbol{\theta})$ evaluated in $\hat{\boldsymbol{\theta}}$. The same $\mathbf{C}_{\mathbf{e}}$ is used as in the calibration method C. Now, we can see that the results provided by the methods A and B do not differ significantly in standard deviation, but in method C, where individual weighting values were used, the precision is much better. For all methods, the theoretical value is still quite far. The main reason for this is that the covariance matrix $\mathbf{C}_{\mathbf{e}}$ was not known precisely in advance, and it had to be estimated first. As a result, the camera parameters are not optimal, and on the other hand, $\mathbf{C}_{\mathbf{e}}$ also became underestimated in this case, causing the theoretical values to be clearly smaller than their true values.

Computationally, the new procedure does not increase the number of the iterations required. For methods A and B, 11 iterations were performed on average. For method C, around 20 iterations were needed in order to achieved convergence. For methods B and C, where the circular geometry is regarded, the average fitting residual for the synthetic images was around 0.01 pixels, which is better than the original requirement of $1/50$ pixels. For method A, the error is over 0.02 pixels, which exceeds the limit.

In reality, the error sources that are not considered in the previous simulations can increase the residual significantly. For example, using the real image shown in Fig. 2a the average residual error is 0.048 pixels (horizontal) and 0.038 pixels (vertical), which is a sign of the small error components originating from the camera electronics, illumination, optical system, and calibration target. Obviously, the accuracy of the calibration result for real images is strongly coupled with these error sources.

5.2 Calibrating the Reverse Distortion Model

In this experiment, the accuracy of the reverse distortion model was tested with a wide range of the distortion parameters. The third step of the calibration procedure was applied to solve the parameter vector $\hat{\delta}'$ for each $\hat{\delta}$. Next, 10,000 random points were generated inside the image region. These simulated points were then corrected using (5) and distorted again using (12). The difference between the resulting coordinates and the original coordinates was examined with respect to the different values of the distortion parameters. The solid curve in Fig. 3a represents the root mean square error in the image coordinates as a function of the first radial distortion parameter k_1. The range of the parameter should cover most of the commonly used off-the-shelf lenses. It can be seen that the error remains below 0.005 pixels for almost all parameter values. As a reference, results given by two other techniques have also been presented. Method 2 is based on the idea of using the distortion model in a reverse direction, as given by (7), with the exception of estimating the parameter values separately for both directions in order to improve the result. Method 3 is a recursive approach and it is based on (8) with

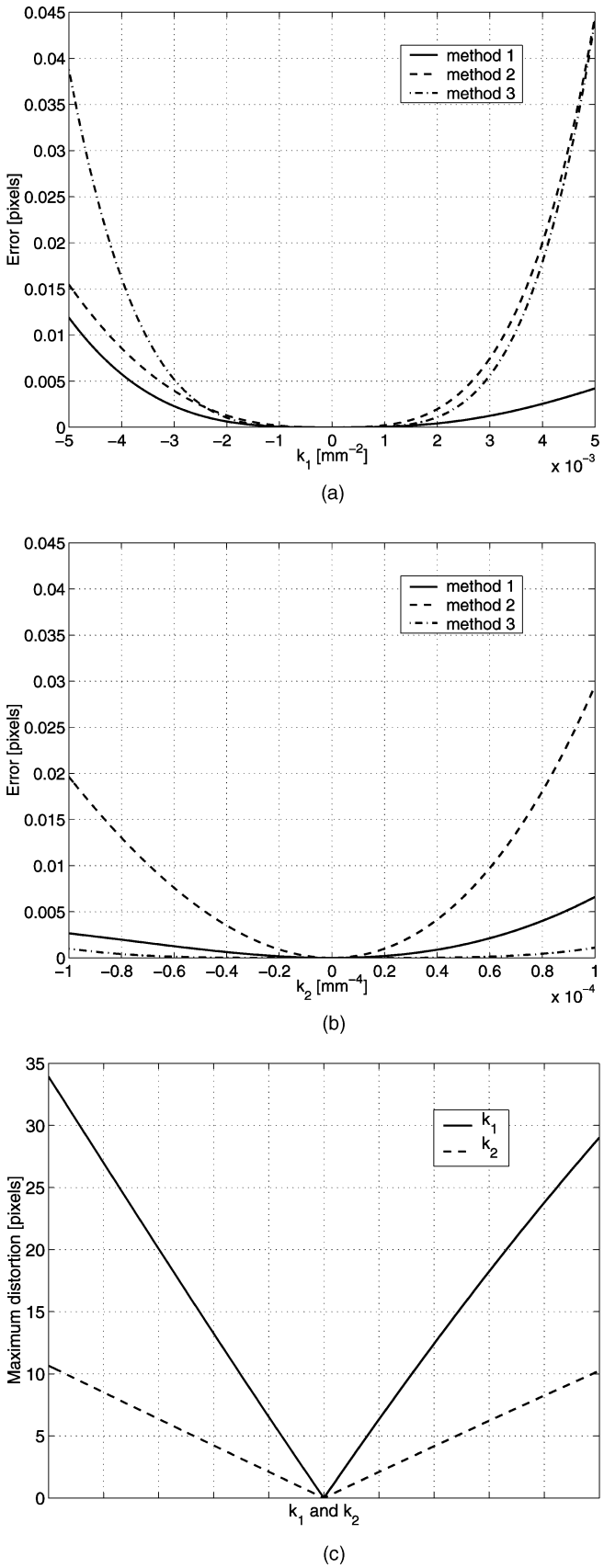


Fig. 3. The error of three different inverse distortion models (a) as a function of k_1, (b) as a function of k_2, and (c) the maximum distortion in the image coordinates for k_1 and k_2.

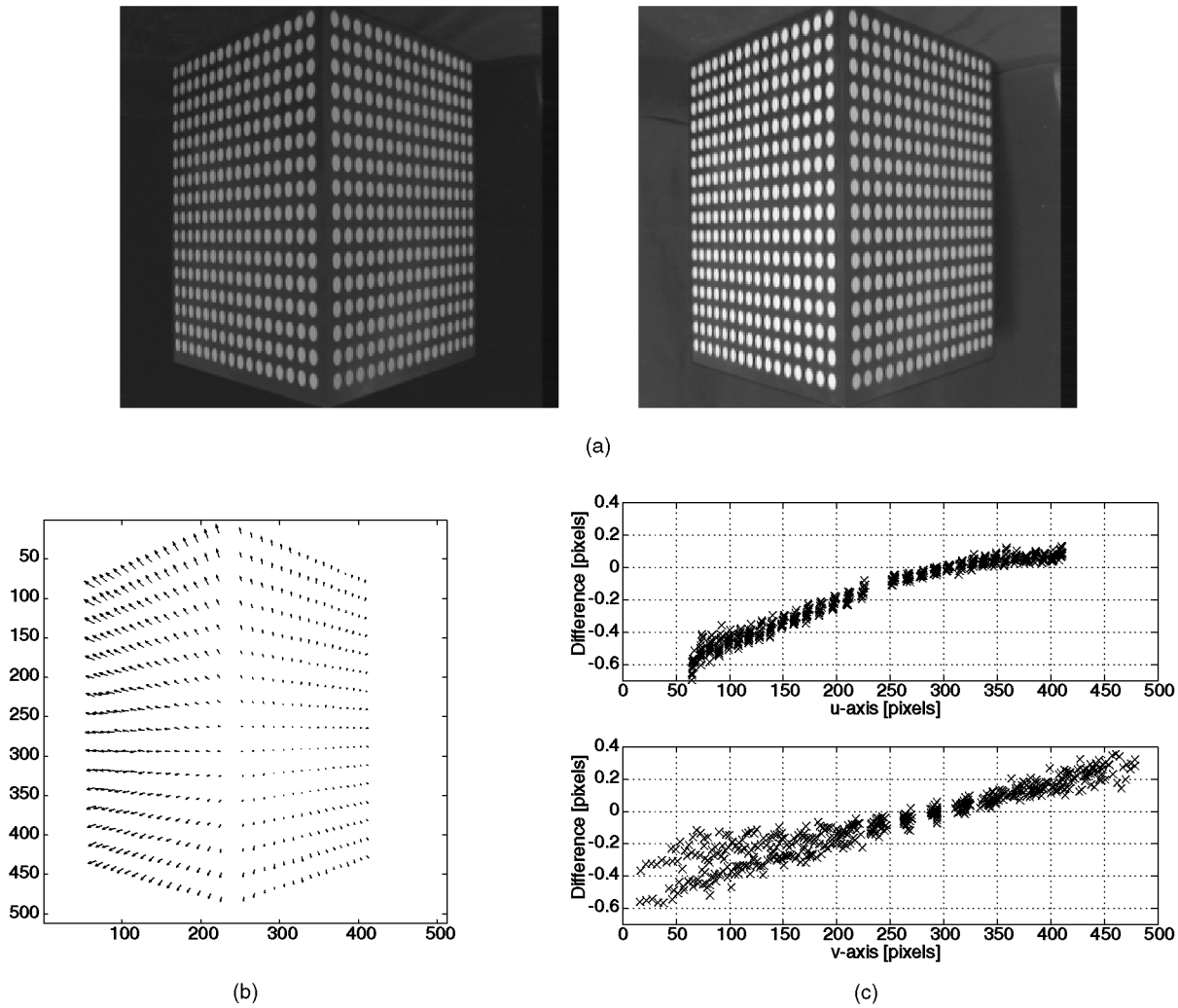


Fig. 4. (a) Calibration object under two different light sources: a fluorescent lamp on the left and a halogen lamp on the right. (b) and (c) Difference ($\mathbf{a}_{halogen}$ - $\mathbf{a}_{fluorescent}$) between detected point locations.

three iterations. These two methods clearly result in a significantly larger error than method 1.

In Fig. 3b, the same test was performed with the second radial distortion coefficient k_2. Now, method 3 seems to outperform method 1, but in both cases the error remains small. Method 2 is clearly the most inaccurate, causing errors that can be significant in precision applications. Fig. 3c shows the maximum radial distortion for both of the two parameters. The first parameter is typically dominating and the effect of the second parameter is much smaller. For some lenses, using the third radial distortion coefficient can be useful, but still its effect remains clearly below the other two parameters. The same test procedure was also performed with decentering distortion, but the error after correcting and distorting the points was noticed to be insignificantly small for all three methods.

6 Other Calibration Errors

The synthetic images used in the previous section were only subject to quantization noise and random Gaussian noise. As noticed with the real image, there are also other sources that may deteriorate the performance of the calibration procedure. In this section, we will briefly review some of the more significant error sources.

Insufficient projection model. Although the distortion model of (6) is derived using exact ray tracing, there can still exist distortion components, especially in wide angle optics that are not compensated for by this model. The camera model used assumes that the principal point coincides with the center of distortion. As stated in [16], this assumption is not eligible for real lens systems. Furthermore, in the pinhole camera model, the rays coming from different directions and distances go through a single point, i.e., the projection center. This is a good approximation for most of the applications, but unfortunately real optics do not behave exactly like that. There is usually some nonlinear dependence between the focal length, imaging distance and the size of the aperture. As a consequence, for apertures larger than a pinhole the focal length is slightly changing, especially in short distances.

Illumination changes. Changes in the irradiance and the wavelength perceived by the camera are typically neglected in geometric camera calibration. However,

variation in lighting conditions can also have a substantial effect on the calibration results. In Fig. 4a, there are two images captured from exactly the same position and angle using the same camera settings, and only the overall lighting has changed from fluorescent to halogen light. Fig. 4b shows the difference between the detected control point locations in the image plane. The difference is smallest near the image center and increases radially when approaching the image borders. From Fig. 4c, we notice that the difference is almost linear as a function of the horizontal and vertical coordinates. The reason for this is chromatic aberration. According to Smith and Thomson [17], chromatic aberration is a measurement of the spread of an image point over a range of colors. It may be represented either as a longitudinal movement of an image plane, or as a change in magnification, but basically it is a result of the dependence of the power of a refracting surface on wavelength.

Camera electronics. Another type of systematic error is distinguished from Fig. 4c, which is the horizontal shift in the observed point locations. This shift is caused by a small change in the overall intensity of lighting. The problem originates from the phase locked loop (PLL) operation, which is affected by the changes in the intensity level of the incoming light. According to Beyer [18], detection of sync pulses is level sensitive and influenced by any changes on the signal level. This deviation is called line jitter, and it can be systematic as in Fig. 4c, or random, which is detected as increased observation noise in the horizontal image coordinates of the control points.

Calibration target. In the calibration procedure presented, it is assumed that the coordinates of the 3D control points are known with such precision that their errors are not observable from the images. If this assumption is not met, the estimates of the camera parameters become inaccurate. The relative accuracy in the object space should be better that the accuracy aspired to in the image space. For example, if the goal is to achieve an accuracy of 1/50 of the pixel or around 1/50,000 of the image size, and if the dimensions of the calibration target are 100 mm by 100 mm in lateral directions, the 3D coordinates of the circular control points should be known to a resolution better than 2μm. Especially systematic errors that exceed the limit can cause significant bias to the parameter estimates.

The effects of the error sources listed above are mixed in the observations and it is often very difficult to recognize individual sources. However, the mixed effect of those sources that are not compensated for by the camera model can be observed from the fitting residual as increased random noise and as a systematic fluctuation. In [7], the systematic component of the remaining error is called *anomalous distortion*. In precise camera calibration, this component should be so small that its influence on the parameter estimates becomes insignificant.

7 Conclusions

Geometric camera calibration is needed to describe the mapping between 3D world coordinates and 2D image coordinates. An optimal calibration technique should produce unbiased and minimum variance estimates of the camera parameters. In practice, this is quite difficult to achieve due to different error sources affecting the imaging process. The calibration procedure suggested in this paper utilizes circular control points and performs mapping from world coordinates into image coordinates and backward from image coordinates to lines of sight or 3D plane coordinates. The camera model used allows least-squares optimization with the distorted image coordinates. The amount of computation is slightly increased with respect to the traditional camera model, but the number of the iterations needed in optimization remains the same if weighting is not used. In the weighted least squares, the number of the iterations is doubled. Experiments showed that an accuracy of 1/50 of the pixel size is achievable with this technique if error sources, such as line jitter and chromatic aberration, are eliminated.

The camera calibration toolbox for Matlab used in the experiments is available on the Internet at http://www.ee.oulu.fi/~jth/calibr/.

Acknowledgments

The author would like to thank the anonymous reviewers for valuable comments and suggestions. The financial support of the Academy of Finland and the Graduate School in Electronics, Telecommunications, and Automation (GETA) is gratefully acknowledged.

References

[1] D.C. Brown, "Evolution, Application and Potential of the Bundle Method of Photogrammetric Triangulation," *Proc. ISP Symp.*, pp. 69, Sept. 1974.
[2] 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.
[3] Y.I. Abdel-Aziz and H.M. Karara, "Direct Linear Transformation into Object Space Coordinates in Close-Range Photogrammetry," *Proc. Symp. Close-Range Photogrammetry*, pp. 1-18, Jan. 1971.
[4] S. Ganapathy, "Decomposition of Transformation Matrices for Robot Vision," *Pattern Recognition Letters*, vol. 2, pp. 401-412, 1984.
[5] O.D. Faugeras and G. Toscani, "Camera Calibration for 3D Computer Vision," *Proc. Int'l Workshop Industrial Applications of Machine Vision and Machine Intelligence*, pp. 240-247, Feb. 1987.
[6] T. Melen, "Geometrical Modelling and Calibration of Video Cameras for Underwater Navigation," doctoral dissertation, Norwegian Univ. of Science and Technology, Trondheim, Norway, 1994.
[7] *Manual of Photogrammetry*, fourth ed., C.C. Slama, ed., Falls Church, Va.: Am. Soc. Photogrammetry, 1980.
[8] H. Haggrén, "Photogrammetric Machine vision," *Optics and Lasers in Eng.*, vol. 10, nos. 3-4, pp. 265-286, 1989.
[9] R.K. Lenz and R.Y. Tsai, "Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Metrology," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 713-720, May 1988.
[10] J. Weng, P. Cohen, and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, nol. 14, no. 10, pp. 965-980, Oct. 1992.

[11] K. Kanatani, *Geometric Computation for Machine Vision.* Oxford: Clarendon Press, 1993.
[12] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C—The Art of Scientific Computing,* second ed., Cambridge Univ. Press, 1992.
[13] G. Strang, *Linear Algebra and Its Applications,* third ed., San Diego, Calif.: Harcourt Brace Jovanovich, 1988.
[14] J. Heikkilä, "Moment and Curvature Preserving Technique for Accurate Ellipse Boundary Detection," *Proc. 14th Int'l Conf. Pattern Recognition,* pp. 734-737, Aug. 1998.
[15] K. Kanatani, "Statistical Bias of Conic Fitting and Renormalization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 3, pp. 320-326, Mar. 1994.
[16] R.G. Willson and S.A. Shafer, "What is the Center of the Image?," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 670-671, 1993.
[17] F.G. Smith and J.H. Thomson, *Optics,* second ed., Wiley, Chichester, 1988.
[18] H.A. Beyer, "Linejitter and Geometric Calibration of CCD Cameras," *ISPRS J. Photogrammetry and Remote Sensing,* vol. 45, pp. 17-32, 1990.

Janne Heikkilä received the MSc degree in electrical engineering from the University of Oulu, Finland, in 1993 and the PhD degree in information engineering in 1998. From September 1998 to July 2000, he was a postdoctoral fellow in Machine Vision and Media Processing Unit at the University of Oulu. Currently, he is an acting professor of signal processing engineering at the University of Oulu. His research interests include camera-based precise 3D measurements, geometric camera calibration, human tracking from image sequences, and motion estimation.