A Four-step Camera Calibration Procedure with Implicit Image Correction

Janne Heikkilä and Olli Silvén
Infotech Oulu and Department of Electrical Engineering
University of Oulu
FIN-90570 Oulu, Finland
email: jth@ee.oulu.fi, olli@ee.oulu.fi

Abstract

In geometrical camera calibration the objective is to determine a set of camera parameters that describe the mapping between 3-D reference coordinates and 2-D image coordinates. Various methods for camera calibration can be found from the literature. However, surprisingly little attention has been paid to the whole calibration procedure, i.e., control point extraction from images, model fitting, image correction, and errors originating in these stages. The main interest has been in model fitting, although the other stages are also important. In this paper we present a four-step calibration procedure that is an extension to the two-step method. There is an additional step to compensate for distortion caused by circular features, and a step for correcting the distorted image coordinates. The image correction is performed with an empirical inverse model that accurately compensates for radial and tangential distortions. Finally, a linear method for solving the parameters of the inverse model is presented.

1. Introduction

Camera calibration in the context of three-dimensional machine vision is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3-D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters) [8]. In many cases, the overall performance of the machine vision system strongly depends on the accuracy of the camera calibration.

Several methods for geometric camera calibration are presented in the literature. The classic approach [7] that originates from the field of photogrammetry solves the problem by minimizing a nonlinear error function. Due to slowness and computational burden of this technique, closed-form solutions have been also suggested (e.g. [8],[1],[5]). However, these methods are based on certain simplifications in the camera model, and therefore, they do not provide as good results as nonlinear minimization. There are also calibration procedures where both nonlinear minimization and a closed form solution are used (e.g. [5],[10]). In these two-step methods, the initial parameter values are computed linearly and the final values are obtained with nonlinear minimization. The methods where the camera model is based on physical parameters, like focal length and principal point, are called explicit methods. In most cases, the values for these parameters are in themselves useless, because only the relationship between 3-D reference coordinates and 2-D image coordinates is required. In implicit camera calibration, the physical parameters are replaced by a set of non-physical implicit parameters that are used to interpolate between some known tie-points (e.g. [9]).

In this paper, we present a four-step calibration procedure that is an extension to the two-step procedure. Section 2.1. describes the closed-form solution to the problem using a direct linear transformation (DLT). Section 2.2. briefly discuss the nonlinear parameter estimation. The third step is needed if we use control points whose projections are larger than one pixel in size. In Section 2.3., we only consider circular features, but similar analysis can be made for arbitrary feature shapes. There are also other error sources in feature extraction, like changes in the illumination, but they are discussed in [4]. The fourth step of the procedure is presented in Section 3. and it solves the image correction problem. Image correction is performed by using a new implicit model that interpolates the correct image points based on the physical camera parameters derived in previous steps. A complete Matlab toolbox for performing this calibration procedure will be available through the Internet.

2. Explicit camera calibration

Physical camera parameters are commonly divided into extrinsic and intrinsic parameters. Extrinsic parameters are needed to transform object coordinates to a camera centered coordinate frame. In multi-camera systems, the extrinsic parameters also describe the relationship between the cameras. The pinhole camera model is based on the principle of collinearity, where each point in the object space is projected by a straight line through the projection center into the image plane. The origin of the camera coor-

1063-6919/97 $10.00

dinate system is in the projection center at the location (X_0, Y_0, Z_0) with respect to the object coordinate system, and the z-axis of the camera frame is perpendicular to the image plane. The rotation is represented using Euler angles ω, φ, and κ that define a sequence of three elementary rotations around x, y, z-axis respectively. The rotations are performed clockwise, first around the x-axis, then the y-axis that is already once rotated, and finally around the z-axis that is twice rotated during the previous stages.

In order to express an arbitrary object point P at location (X_i, Y_i, Z_i) in image coordinates, we first need to transform it to camera coordinates (x_i, y_i, z_i). This transformation consists of a translation and a rotation, and it can be performed by using the following matrix equation:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (1)$$

where

$$m_{12}=\sin\omega\sin\varphi\cos\kappa-\cos\omega\sin\kappa \qquad m_{11}=\cos\varphi\cos\kappa$$
$$m_{22}=\sin\omega\sin\varphi\sin\kappa+\cos\omega\cos\kappa \qquad m_{21}=\cos\varphi\sin\kappa$$
$$m_{13}=\cos\omega\sin\varphi\cos\kappa+\sin\omega\sin\kappa \qquad m_{31}=-\sin\varphi$$
$$m_{23}=\cos\omega\sin\varphi\sin\kappa-\sin\omega\cos\kappa \qquad m_{32}=\sin\omega\cos\varphi$$
$$m_{33}=\cos\omega\cos\varphi \qquad x_0=-m_{11}X_0-m_{12}Y_0-m_{13}Y_0$$
$$y_0=-m_{21}X_0-m_{22}Y_0-m_{23}Y_0 \qquad z_0=-m_{31}X_0-m_{32}Y_0-m_{33}Y_0$$

The intrinsic camera parameters usually include the effective focal length f, scale factor s_u, and the image center (u_0, v_0) also called the principal point. Here, as usual in computer vision literature, the origin of the image coordinate system is in the upper left corner of the image array. The unit of the image coordinates is pixels, and therefore coefficients D_u and D_v are needed to change the metric units to pixels. These coefficients can be typically obtained from the data sheets of the camera and framegrabber. In fact, their precise values are not necessary, because they are linearly dependent on the focal length f and the scale factor s_x. By using the pinhole model, the projection of the point (x_i, y_i, z_i) to the image plane is expressed as

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{f}{z_i} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2)$$

The corresponding image coordinates (u_i', v_i') in pixels are obtained from the projection ($\tilde{u}_i$, $\tilde{v}_i$) by applying the following transformation:

$$\begin{bmatrix} u_i' \\ v_i' \end{bmatrix} = \begin{bmatrix} D_u s_u \tilde{u}_i \\ D_v \tilde{v}_i \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad (3)$$

The pinhole model is only an approximation of the real camera projection. It is a useful model that enables simple mathematical formulation for the relationship between object and image coordinates. However, it is not valid when high accuracy is required and therefore, a more comprehensive camera model must be used. Usually, the pinhole model is a basis that is extended with some corrections for the systematically distorted image coordinates. The most commonly used correction is for the radial lens distortion that causes the actual image point to be displaced radially in the image plane [7]. The radial distortion can be approximated using the following expression:

$$\begin{bmatrix} \delta u_i^{(r)} \\ \delta v_i^{(r)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_i(k_1 r_i^2 + k_2 r_i^4 + \ldots) \\ \tilde{v}_i(k_1 r_i^2 + k_2 r_i^4 + \ldots) \end{bmatrix} \quad (4)$$

where k_1, k_2,... are coefficients for radial distortion, and $r_i = \sqrt{\tilde{u}_i^2 + \tilde{v}_i^2}$. Typically, one or two coefficients are enough to compensate for the distortion.

Centers of curvature of lens surfaces are not always strictly collinear. This introduces another common distortion type, decentering distortion which has both a radial and tangential component [7]. The expression for the tangential distortion is often written in the following form:

$$\begin{bmatrix} \delta u_i^{(t)} \\ \delta v_i^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1\tilde{u}_i\tilde{v}_i + p_2(r_i^2 + 2\tilde{u}_i^2) \\ p_1(r_i^2 + 2\tilde{v}_i^2) + 2p_2\tilde{u}_i\tilde{v}_i \end{bmatrix} \quad (5)$$

where p_1 and p_2 are coefficients for tangential distortion.

Other distortion types have also been proposed in the literature. For example, Melen [5] uses the correction term for linear distortion. This term is relevant if the image axes are not orthogonal. In most cases the error is small and the distortion component is insignificant. Another error component is thin prism distortion. It arises from imperfect lens design and manufacturing, as well as camera assembly. This type of distortion can be adequately modelled by the adjunction of a thin prism to the optical system, causing additional amounts of radial and tangential distortions [2],[10].

A proper camera model for accurate calibration can be derived by combining the pinhole model with the correction for the radial and tangential distortion components:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_u s_u(\tilde{u}_i + \delta u_i^{(r)} + \delta u_i^{(t)}) \\ D_v(\tilde{v}_i + \delta v_i^{(r)} + \delta v_i^{(t)}) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad (6)$$

In this model the set of intrinsic parameters (f, s_u, u_0, v_0) is augmented with the distortion coefficients k_1,..., k_n, p_1 and p_2. These parameters are also known as physical camera parameters, since they have a certain physical meaning. Generally, the objective of the explicit camera calibration procedure is to determine optimal values for these parameters based on image observations of a known 3-D target. In the case of self-calibration the 3-D coordinates of the target points are also included in the set of unknown parameters. However, the calibration procedure presented in this article is performed with a known target.

2.1. Linear parameter estimation

The direct linear transformation (DLT) was originally developed by Abdel-Aziz and Karara [1]. Later, it was revised in several publications, e.g. in [5] and [3].

The DLT method is based on the pinhole camera model (see Eq. (3)), and it ignores the nonlinear radial and tangential distortion components. The calibration procedure consists of two steps. In the first step the linear transformation from the object coordinates (X_i, Y_i, Z_i) to image coordinates (u_i, v_i) is solved. Using a homogeneous 3 x 4 matrix representation for matrix **A** the following equation can be written:

$$\begin{bmatrix} u_i w_i \\ v_i w_i \\ w_i \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{7}$$

We can solve the parameters $a_{11},..., a_{34}$ of the DLT matrix by eliminating w_i. Let us denote

$$\mathbf{L} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1u_1 & -Y_1u_1 & -Z_1u_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1v_1 & -Y_1v_1 & -Z_1v_1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -X_iu_i & -Y_iu_i & -Z_iu_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -X_iv_i & -Y_iv_i & -Z_iv_i & -v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -X_Nu_N & -Y_Nu_N & -Z_Nu_N & -u_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -X_Nv_N & -Y_Nv_N & -Z_Nv_N & -v_N \end{bmatrix}$$

$$\mathbf{a} = [a_{11}, a_{12}, a_{13}, a_{14}, a_{21}, a_{22}, a_{23}, a_{24}, a_{31}, a_{32}, a_{33}, a_{34}]^T$$

The following matrix equation for N control points is obtained [5]:

$$\mathbf{La} = \mathbf{0} \tag{8}$$

By replacing the correct image points (u_i, v_i) with observed values (U_i, V_i) we can estimate the parameters $a_{11},..., a_{34}$ in a least squares fashion. In order to avoid a trivial solution $a_{11},..., a_{34} = 0$, a proper normalization must be applied. Abdel-Aziz and Karara [1] used the constraint $a_{34} = 1$. Then, the equation can be solved with a pseudoinverse technique. The problem with this normalization is that a singularity is introduced, if the correct value of a_{34} is close to zero. Instead of $a_{34} = 1$ Faugeras and Toscani [3] suggested the constraint $a_{31}^2 + a_{32}^2 + a_{33}^2 = 1$ which is singularity free.

The parameters $a_{11},..., a_{34}$ do not have any physical meaning, and thus the first step where their values are estimated can be also considered as the implicit camera calibration stage. There are techniques for extracting some of the physical camera parameters from the DLT matrix, but not many are able to solve all of them. Melen [5] proposed a method based on RQ decomposition where a set of eleven physical camera parameters are extracted from the DLT matrix. The decomposition is as follows:

$$\mathbf{A} = \lambda \mathbf{V}^{-1}\mathbf{B}^{-1}\mathbf{FMT} \tag{9}$$

where λ is an overall scaling factor and the matrices **M** and **T** define the rotation and translation from the object coordinate system to the camera coordinate system (see Eq. (1)). Matrices **V**, **B**, and **F** contain the focal length f, principal point (u_0, v_0) and coefficients for the linear distortion (b_1, b_2):

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1+b_1 & b_2 & 0 \\ b_2 & 1-b_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The linear distortion correction is used here to compensate for the orthogonality errors of the image coordinate axes. A five step algorithm for solving the parameters is given in [5] and it not represented here. In this procedure, the scale factor s_u is assumed to be 1. In the case of coplanar control point structure, the 3 x 4 DLT matrix becomes singular. Thus, a 3 x 3 matrix with nine unknown parameters must be used. Melen also proposed a method for decomposing the 3 x 3 matrix, but only a subset of physical camera parameters can be estimated.

2.2. Nonlinear estimation

Since no iterations are required, direct methods are computationally fast. However, they have at least the following two disadvantages. First, lens distortion cannot be incorporated, and therefore, distortion effects are not generally corrected, although some solutions also for this problem have been presented. For example, Shih *et al.* [6] used a method where the estimation of the radial lens distortion coefficient is transformed into an eigenvalue problem. The second disadvantage of linear methods is more difficult to be fixed. Since, due to the objective to construct a noniterative algorithm, the actual constraints in the intermediate parameters are not considered. Consequently, in the presence of noise, the intermediate solution does not satisfy the constraints, and the accuracy of the final solution is relatively poor [10]. Due to these difficulties the calibration results obtained in Section 2.1. are not accurate enough.

With real cameras the image observations are always contaminated by noise. As we know, there are various error components incorporated in the measurement process, but these error components are discussed more profoundly in [4]. If the systematic parts of the measurement error are compensated for, it is convenient to assume that the error is white Gaussian noise. Then, the best estimate for the camera parameters can be obtained by minimizing the residual between the model and N observations (U_i, V_i), where $i = 1,..., N$. In the case of Gaussian noise, the objective function is expressed as a sum of squared residuals:

$$F = \sum_{i=1}^{N} (U_i - u_i)^2 + \sum_{i=1}^{N} (V_i - v_i)^2 \quad (10)$$

The least squares estimation technique can be used to minimize Eq. (10). Due to the nonlinear nature of the camera model, simultaneous estimation of the parameters involves applying an iterative algorithm. For this problem the Levenberg-Marquardt optimization method has been shown to provide the fastest convergence. However, without proper initial parameter values the optimization may stick in a local minimum and thereby cause the calibration to fail. This problem can be avoided by using the parameters from the DLT method as the initial values for the optimization. A global minimum of Eq. (10) is then usually achieved after a few iterations.

Two coefficients for both radial and tangential distortion is normally enough [4]. Our experiments have also shown that the linear distortion in modern CCD arrays is typically negligible. Thus, the parameters b_1, b_2 can be usually left out, and totally eight intrinsic parameters are then estimated. The number of extrinsic parameters depends on the number of camera views. Using a 3-D target structure, only a single viewpoint is required. In the case of a coplanar target, a singularity is introduced that limits the number of parameters that can be estimated from a single view. Therefore, multiple views are required in order to solve all the intrinsic parameters. The number of extrinsic parameters is now added by six for each perspective view.

2.3. Correction for the asymmetric projection

Perspective projection is generally not a shape preserving transformation. Only lines are mapped as lines on the image plane. Two- and three-dimensional objects with a non-zero projection area are distorted if they are not coplanar with the image plane. This is true for arbitrary shaped features, but in this article we are only concerned with circles, because of their simple analytic formulation. Another reason is that they are very common shapes in many man-made objects.

The center points of the circles are often located from the images with subpixel precision, but the distortion caused by the perspective projection is not typically considered. Perspective projection distorts the shape of the circular features in the image plane depending on the angle and displacement between the object surface and the image plane. Only when the surface and the image plane are parallel, projections remain circular. These facts are well-known, but the mathematical formulation of the problem has been often disregarded. Therefore, we shall next review the necessary equations.

Let the coordinate system Ω_1 $(X, Y, Z) \in \Re^3$ be centered in the camera focus O, and let its Z-axis be perpendicular to the object surface Π_1 (see Fig. 1). The rays coming from

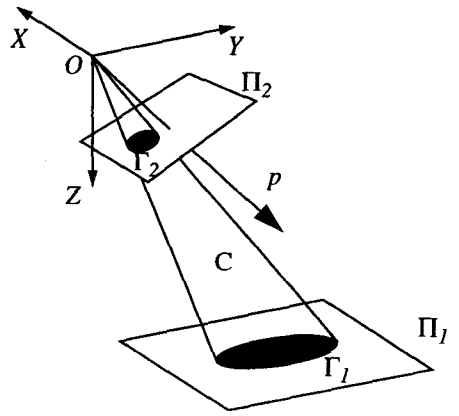

Figure 1. Perspective projection of a circle.

the circle Γ_1 that is located on the surface Π_1 form a skewed cone, whose boundary curve C can be expressed as follows:

$$(X - \alpha Z)^2 + (Y - \beta Z)^2 = \gamma^2 Z^2 \quad (11)$$

Parameters α and β specify the skewness of the cone in X and Y directions and the parameter γ specifies the sharpness of the cone. Thus, if the distance from the camera focus to the object surface is denoted by d, the circle equation becomes $(X - \alpha d)^2 + (Y - \beta d)^2 = (\gamma d)^2$.

The camera coordinate system Ω_2 $(x, y, z) \in \Re^3$ is also centered in the camera focus, but its z-axis is orthogonal to the image plane Π_2, and its x- and y-axes are parallel to the image axes u and v. Thus, the transformation from Ω_2 to Ω_1 is expressed by using the following rotation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (12)$$

where the vectors $[a_{11}, a_{21}, a_{31}]^T$, $[a_{12}, a_{22}, a_{32}]^T$, and $[a_{13}, a_{23}, a_{33}]^T$ form an orthonormal basis. Now, we can express Eq. (11) in camera coordinates

$$\begin{aligned} &[(a_{11} - \alpha a_{31})x + (a_{12} - \alpha a_{32})y + (a_{13} - \alpha a_{33})z]^2 \\ &+ [(a_{21} - \beta a_{31})x + (a_{22} - \beta a_{32})y + (a_{23} - \beta a_{33})z]^2 \\ &= \gamma^2 (a_{31}x + a_{32}y + a_{33}z)^2 \end{aligned} \quad (13)$$

Let us denote the focal length, i.e. the orthogonal distance between O and Π_2, by f. Then, the intersection Γ_2 of C and Π_2 is expressed as:

$$\begin{aligned} &(n^2 + k^2 - r^2)x^2 + 2(kl + np - rs)xy + (l^2 + p^2 - s^2)y^2 \\ &+ 2(km + nq - rt)x + 2(lm + pq - st) + m^2 + q^2 - t^2 = 0 \end{aligned} \quad (14)$$

where

$$\begin{aligned} k &= a_{11} - ta_{31} & n &= a_{21} - sa_{31} & r &= \gamma a_{31} \\ l &= a_{12} - ta_{32} & p &= a_{22} - sa_{32} & s &= \gamma a_{32} \\ m &= (a_{13} - ta_{33})f & q &= (a_{23} - sa_{33})f & t &= \gamma a_{33} f \end{aligned}$$

We notice from Eq. (14) that the projection is a quadratic curve and its geometrical interpretation can be a circle, hyperbola, parabola, or ellipse. In practice, due to the limited

field of view the projection will be a circle or ellipse.

From Eq. (14) the center of the ellipse $(\tilde{u}_c, \tilde{v}_c)$ can be expressed as

$$\tilde{u}_c = \frac{(kp-nl)(lq-pm)-(ks-lr)(tl-ms)-(ns-pr)(wp-qs)}{(kp-nl)^2-(ks-lr)^2-(ns-pr)^2} \quad (15)$$

$$\tilde{v}_c = \frac{(kp-nl)(mn-kq)-(ks-lr)(mr-kt)-(ns-pr)(qr-nt)}{(kp-nl)^2-(ks-lr)^2-(ns-pr)^2}$$

In order to find out what is the projection of the circle center, let us consider a situation where the radius of the circle is zero, i.e. $\gamma = 0$. Consequently, r, s, and t become zero, and we obtain the position of the projected point that is due to the symmetry of the circle also the projection of the circle center $(\tilde{u}_0, \tilde{v}_0)$:

$$\tilde{u}_0 = (lq-pm)/(kp-nl) \quad \tilde{v}_0 = (mn-kq)/(kp-nl) \quad (16)$$

For non-zero radius ($\gamma > 0$) there are only some special cases when Eqs (15) and (16) are equal, e.g. the rotation is performed around the Z-axis ($a_{31} = a_{32} = 0$). Generally, we can state that the ellipse center and projected circle center are not the same for circular features with non-zero radius.

Ellipse fitting or the center of gravity method produces estimates of the ellipse center. However, what we usually want to know is the projection of the circle center. As a consequence of the previous discussion, we notice that the location is biased and it should be corrected using Eqs (15) and (16). Especially, in camera calibration this is very important, because the circular dot patterns are usually viewed in skew angles.

There are at least two possibilities to correct this projection error. The first solution is to include the correction $(\tilde{u}_c - \tilde{u}_0, \tilde{v}_c - \tilde{v}_0)$ to the camera model. An optimal estimate in a least squares sense is then obtained. However, this solution degrades the convergence rate considerably, and thus increases the amount of computation. Another possibility is to compute the camera parameters recursively, when the parameters obtained in the least squares estimation step are used to evaluate Eqs (15) and (16). Observed image coordinates (U_i, V_i) are then corrected with the following formula:

$$\begin{aligned} U_i' &= U_i - D_u s_u(\tilde{u}_{c,i} - \tilde{u}_{0,i}) \\ V_i' &= V_i - D_v(\tilde{v}_{c,i} - \tilde{v}_{0,i}) \end{aligned} \quad (17)$$

After correction, the camera parameters are recomputed. The parameters are not optimal in a least squares sense, but the remaining error is so small that no further iterations are needed.

The significance of the third calibration step is demonstrated in Fig. 2 a) with an image of a cubic 3-D calibration object. Since the two visible surfaces of the object are perpendicular there is no way to select the viewing angle so that the projection asymmetry vanishes. Fig. 2 b) shows the error in horizontal and vertical directions. The error in this case is quite small (about 0.14 pixels peak to peak), but it is systematic causing bias to the camera parameters.

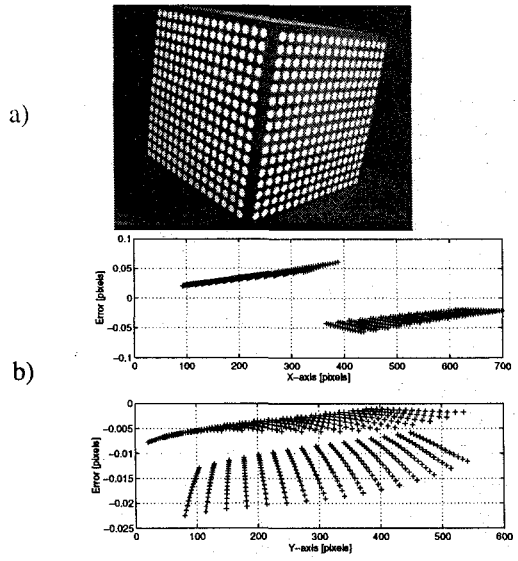

Figure 2. a) A view of the calibration object. b) Error caused by the asymmetrical dot projection.

3. Image correction

The camera model given in Eq. (6) expresses the projection of the 3-D points on the image plane. However, it does not give a direct solution to the back-projection problem, in which we want to recover the line of sight from image coordinates. If both radial and tangential distortion components are considered, we can notice that there is no analytic solution to the inverse mapping. For example, two coefficients for radial distortion cause the camera model in Eq. (6) to become a fifth order polynomial:

$$\begin{aligned} u_i &= D_u s_u(k_2\tilde{u}_i^5 + 2k_2\tilde{u}_i^3\tilde{v}_i^2 + k_2\tilde{u}_i\tilde{v}_i^4 + k_1\tilde{u}_i^3 + k_1\tilde{u}_i\tilde{v}_i^2 \\ &\quad + 3p_2\tilde{u}_i^2 + 2p_1\tilde{u}_i\tilde{v}_i + p_2\tilde{v}_i^2 + \tilde{u}_i) + u_0 \\ v_i &= D_v(k_2\tilde{u}_i^4\tilde{v}_i + 2k_2\tilde{u}_i^2\tilde{v}_i^3 + k_2\tilde{v}_i^5 + k_1\tilde{u}_i^2\tilde{v}_i + k_1\tilde{v}_i^3 \\ &\quad + p_1\tilde{u}_i^2 + 2p_2\tilde{u}_i\tilde{v}_i + 3p_1\tilde{v}_i^2 + \tilde{v}_i) + v_0 \end{aligned} \quad (18)$$

We can infer from Eq. (18) that a nonlinear search is required to recover $(\tilde{u}_i, \tilde{v}_i)$ from (u_i, v_i). Another alternative is to approximate the inverse mapping. Only few solutions to the back-projection problem can be found from the literature, although the problem is evident in many applications. Melen [5] used an iterative approach to estimate the undistorted image coordinates. He proposed the following two-iteration process:

$$\mathbf{q}_i' = \mathbf{q}_i'' - \delta(\mathbf{q}_i'' - \delta(\mathbf{q}_i'')) \quad (19)$$

where vectors $\mathbf{q}_i''$ and $\mathbf{q}_i'$ contain the distorted and the corrected image coordinates respectively. The function $\delta(\mathbf{q})$ represents the distortion in image location $\mathbf{q}$. In our tests this method gave a maximum residual of about 0.1 pixels for typical lens distortion parameters. This may be enough for some applications, but if better accuracy is needed then

more iterations should be accomplished.

A few implicit methods e.g. a two-plane method as proposed by Wei and Ma [9] solve the back-projection problem by determining a set of non-physical or implicit parameters to compensate for the distortion. Due to a large number of unknown parameters, this technique requires a dense grid of observations from the whole image plane in order to become accurate. However, if we know the physical camera parameters based on explicit calibration, it is possible to solve the unknown parameters by generating a dense grid of points $(\tilde{u}_i, \tilde{v}_i)$ and calculating the corresponding distorted image coordinates (u_i, v_i) by using the camera model in Eq. (6). Based on the implicit camera model proposed by Wei and Ma [9] we can express the mapping from (u_i, v_i) to $(\tilde{u}_i, \tilde{v}_i)$ as follows:

$$\tilde{u}_i = \frac{\sum_{0 \le j+k \le N} a_{jk}^{(1)} u_i^j v_i^k}{\sum_{0 \le j+k \le N} a_{jk}^{(3)} u_i^j v_i^k} \qquad \tilde{v}_i = \frac{\sum_{0 \le j+k \le N} a_{jk}^{(2)} u_i^j v_i^k}{\sum_{0 \le j+k \le N} a_{jk}^{(3)} u_i^j v_i^k} \tag{20}$$

Wei and Ma used third order polynomials in their experiments. In our tests, we noticed that it only provides about 0.1 pixel accuracy with typical camera parameters. This is quite clear, since we have a camera model that contains fifth order terms (see Eq. (18)). Thus, at least fifth order approximations should be applied. This leads to equations where each set of unknown parameters $\{a_{jk}^{(n)}\}$ includes 21 terms. It can be expected that there are also redundant parameters that may be eliminated. After thorough simulations, it was found that the following expression compensated for the distortions so that the maximum residual error was less than 0.01 pixel units, even with a substantial amount of distortion present:

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{1}{G} \begin{bmatrix} \tilde{u}_i' + \tilde{u}_i'(a_1 r_i^2 + a_2 r_i^4) + 2a_3 \tilde{u}_i' \tilde{v}_i' + a_4(r_i^2 + 2\tilde{u}_i'^2) \\ \tilde{v}_i' + \tilde{v}_i'(a_1 r_i^2 + a_2 r_i^4) + a_3(r_i^2 + 2\tilde{v}_i'^2) + 2a_4 \tilde{u}_i' \tilde{v}_i' \end{bmatrix} \tag{21}$$

and

$$G = (a_5 r_i^2 + a_6 \tilde{u}_i' + a_7 \tilde{v}_i' + a_8) r_i^2 + 1 \tag{22}$$

where $\tilde{u}_i' = (u_i - u_0)/(D_u s_u)$, $\tilde{v}_i' = (v_i - v_0)/D_v$, and $r_i = \sqrt{\tilde{u}_i'^2 + \tilde{v}_i'^2}$. If we compare this implicit inverse model to the camera model in Eq. (6) we notice that also the inverse model has components which resemble radial and tangential distortions. The counterparts for the distortion parameters k_1, k_2, p_1, and p_2 are the coefficients a_1,..., a_4.

The model (21)-(22) contains only eight unknown parameters instead of 63 parameters that were in the original fifth-order model in Eq. (20). Back-projection using this model will require less computation than the iterative approach suggested by Melen giving also more accurate results. The parameters a_1,..., a_8 can be solved either iteratively using the least squares technique, when the smallest fitting residual is obtained, or directly, when the result is very close to the optimal.

In order to solve the unknown parameters for the inverse model, N tie-points $(\tilde{u}_i, \tilde{v}_i)$ and $(\tilde{u}_i', \tilde{v}_i')$ covering the whole image area must be generated. In practice, a grid of about 1000 - 2000 points, e.g. 40 x 40, is enough. Let us define

$\mathbf{u}_i =$
$[-\tilde{u}_i' r_i^2, -\tilde{u}_i' r_i^4, -2\tilde{u}_i' \tilde{v}_i', -(r_i^2 + 2\tilde{u}_i'^2), \tilde{u}_i r_i^4, \tilde{u}_i \tilde{u}_i' r_i^2, \tilde{u}_i \tilde{v}_i' r_i^2, \tilde{u}_i r_i^2]^T$

$\mathbf{v}_i =$
$[-\tilde{v}_i' r_i^2, -\tilde{v}_i' r_i^4, -(r_i^2 + 2\tilde{v}_i'^2), -2\tilde{u}_i' \tilde{v}_i', \tilde{v}_i r_i^4, \tilde{v}_i \tilde{u}_i' r_i^2, \tilde{v}_i \tilde{v}_i' r_i^2, \tilde{v}_i r_i^2]^T$

$\mathbf{T} = [\mathbf{u}_1, \mathbf{v}_1, \ldots, \mathbf{u}_i, \mathbf{v}_i, \ldots, \mathbf{u}_N, \mathbf{v}_N]^T$

$\mathbf{p} = [a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8]^T$

$\mathbf{e} =$
$[\tilde{u}_1' - \tilde{u}_1, \tilde{v}_1' - \tilde{v}_1, \ldots, \tilde{u}_i' - \tilde{u}_i, \tilde{v}_i' - \tilde{v}_i, \ldots, \tilde{u}_N' - \tilde{u}_N, \tilde{v}_N' - \tilde{v}_N]^T$

Using Eqs (21) and (22) the following relation is obtained:

$$\mathbf{e} = \mathbf{T}\mathbf{p} \tag{23}$$

The vector $\mathbf{p}$ is now estimated in a least squares sense:

$$\hat{\mathbf{p}} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{e} \tag{24}$$

The parameters computed based on Eq. (24) are used in Eqs (21) and (22) to correct arbitrary image coordinates (u, v). The actual coordinates are then obtained by interpolation based on the generated coordinates $(\tilde{u}_i, \tilde{v}_i)$ and $(\tilde{u}_i', \tilde{v}_i')$.

4. Experiments

Explicit camera calibration experiments are reported in [4]. In this section we concentrate on the fourth step, i.e., the image correction. Let us assume that the first three steps have produced the physical camera parameters listed in Table 1.

s_u	f [mm]	u_0 [pixels]	v_0 [pixels]
1.0039	8.3431	367.6093	305.8503
k_1 [mm^{-2}]	k_2 [mm^{-4}]	p_1 [mm^{-1}]	p_2 [mm^{-1}]
-3.186e-03	4.755e-05	-3.275e-05	-1.565e-05

Table 1. Physical camera parameters.

First, we generate an equally spaced grid (40 x 40) of tie-points $(\tilde{u}_i, \tilde{v}_i)$ that cover the entire image and a small portion outside the effective area so that we can guarantee good results also for the border regions. The corresponding distorted coordinates $(\tilde{u}_i', \tilde{v}_i')$ are obtained by applying Eqs (4) and (5). The parameters a_1,..., a_8 are then solved with the LS method in Eq. (24). The results are given in Table 2, and the fitting residual between the inverse model and the true points is shown in Fig. 3.

a_1	a_2	a_3	a_4
-8.328e-03	1.670e-04	3.269e-06	1.568e-05
a_5	a_6	a_7	a_8
2.202e-04	-1.518e-07	-3.428e-08	-1.151e-02

Table 2. Parameters of the inverse model.

The maximum error in the fitting residual is in this case less than 0.0005 pixels. For more intensive distortion, the

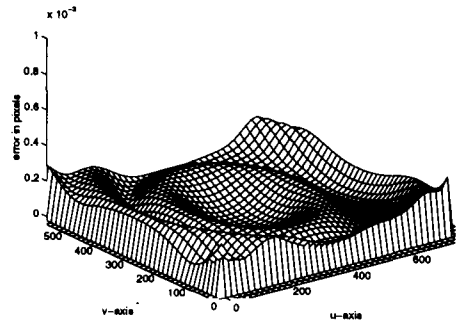

Figure 3. Fitting residual.

error will be slightly bigger, but under realistic conditions always less than 0.01 pixels as the feature detection accuracy (std) was about 0.02 pixels [4].

In the second experiment, we generate a uniformly distributed random set of 2000 points in the image area. These points are first distorted and then corrected with the inverse model. The error originating in this process is represented as histograms in Fig. 4 in both horizontal and vertical directions. The error seems to have the same magnitude as the fitting residual. Therefore, we can affirm that the interpolation between the tie-points does not degrade image correction noticeably.

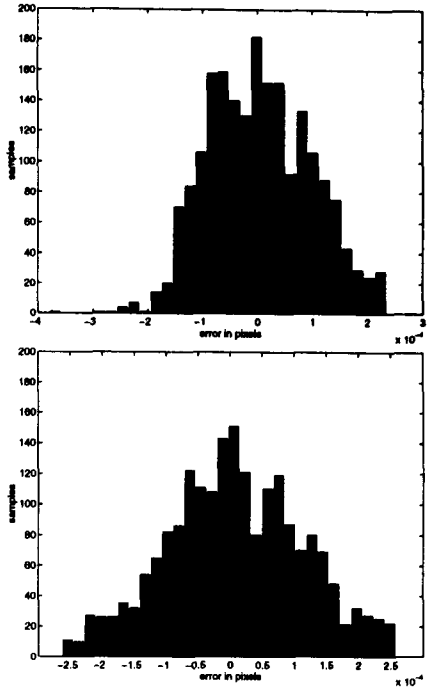

Figure 4. Error caused by the back-projection model for 2000 randomly selected point in horizontal direction and vertical direction.

5. Conclusions

A four-step procedure for camera calibration was presented in this article. This procedure can be utilized in various machine vision applications, but it is most beneficial in camera based 3-D measurements and in robot vision, where high geometrical accuracy is needed. This procedure uses explicit calibration methods for mapping 3-D coordinates to image coordinates and an implicit approach for image correction. The experiments in the last section showed that the error caused by the inverse model is negligible. A Matlab toolbox for performing the calibration procedure is implemented and it will be available through the Internet.

Acknowledgments

The support of The Graduate School in Electronics Telecommunications and Automation (GETA) is gratefully acknowledged.

References

[1] Abdel-Aziz, Y. I. & Karara, H. M. (1971) Direct linear transformation into object space coordinates in close-range photogrammetry. Proc. Symposium on Close-Range Photogrammetry, Urbana, Illinois, p. 1-18.

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

[3] Faugeras, O. D. & Toscani, G. (1987) Camera calibration for 3D computer vision. Proc. International Workshop on Industrial Applications of Machine Vision and Machine Intelligence, Silken, Japan, p. 240-247.

[4] Heikkilä, J. & Silvén, O. (1996) Calibration procedure for short focal length off-the-shelf CCD cameras. Proc. 13th International Conference on Pattern Recognition. Vienna, Austria, p. 166-170.

[5] Melen, T. (1994) Geometrical modelling and calibration of video cameras for underwater navigation. Dr. ing thesis, Norges tekniske høgskole, Institutt for teknisk kybernetikk.

[6] Shih, S. W., Hung, Y. P. & Lin, W. S. (1993) Accurate linear technique for camera calibration considering lens distortion by solving an eigenvalue problem. Optical Engineering 32(1): 138-149.

[7] Slama, C. C. (ed.) (1980) Manual of Photogrammetry, 4th ed., American Society of Photogrammetry, Falls Church, Virginia.

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

[9] Wei, G. Q. & Ma, S. D. (1993) A complete two-plane camera calibration method and experimental comparisons. Proc. 4th International Conference on Computer Vision, Berlin, Germany, p. 439-446.

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