TR-92-001

# Accuracy Assessment On Camera Calibration Method Not Considering Lens Distortion

Sheng-Wen Shih[‡†], Yi-Ping Hung[†] and Wei-Song Lin[‡]

[†]Institute of Information Science, Academia Sinica,
Nankang, Taipei, Taiwan, 11529 R.O.C.

[‡]Institute of Electrical Engineering, National Taiwan University,
Taipei, Taiwan, R.O.C.

# Accuracy Assessment On Camera Calibration Method Not Considering Lens Distortion

Sheng-Wen Shih[‡†], Yi-Ping Hung[†] and Wei-Song Lin[‡]

[†]Institute of Information Science, Academia Sinica,
Nankang, Taipei, Taiwan, 11529 R.O.C.

[‡]Institute of Electrical Engineering, National Taiwan University,
Taipei, Taiwan, R.O.C.

## ABSTRACT

In this paper, calibrating a camera refers to the process of determining the transformation between a 3D object coordinate system and the 2D image coordinate system of the camera. Knowing this transformation, 3D position of a feature point seen in two cameras can be determined by triangulation. Suppose a distortion free lens is had, this 3D–2D transformation can be well approximated by a linear model. Unfortunately, most off–the–shelf lenses have a certain amount of distortion. If our goal is to get highly accurate 3D measurements, we have to consider lens distortion in camera calibration. However, considering lens distortion will not only complicate the camera calibration procedure, but also complicate the subsequent on–line processing such as feature–point correspondence and camera re–calibration. It is hence desirable to avoid considering lens distortion whenever the distortion–induced error is tolerable. This work investigates the effect of neglecting lens distortion, and presents a theoretical analysis of the calibration accuracy. The derived error bound is a function of a few factors including the number of calibration points, the observation error of 2D image points, the radial lens distortion coefficient, the image size and resolution. This error bound provides a guide line for selecting both a proper camera calibration configuration and an appropriate camera model while satisfying the desired accuracy. Experimental results from both computer simulations and real experiments are included in this paper.

# I. INTRODUCTION

To infer 3D objects using two or more images, it is essential to know the relationship between the 2D image coordinate system and the 3D object coordinate system. This relationship can be described by the following two transformations:

(i). Perspective projection of a 3D object point onto a 2D image point -- Given an estimate of a 3D object point and its error covariance, we can predict its projection (mean and covariance) on the 2D image. This is useful for reducing the searching space in matching features between two images, or for hypothesis verification in scene analysis.

(ii). Back projection of a 2D image point to a 3D ray -- Given a 2D image point, there is a ray in the 3D space that the corresponding 3D object point must lie on. If we have two (or more) views available, an estimate of the 3D point location can be obtained by using triangulation. This is useful for inferring 3D information from 2D image features.

The above 3D–2D relationship can be specified by a column vector $\beta$, which contains the geometric camera parameters specifying camera orientation and position, focal length, lens distortion, optical axis misalignment, and pixel size. Determining this 3D–2D relationship, or equivalently, estimating $\beta$, is called (geometric) camera calibration.

The techniques for camera calibration can be classified into two categories: one that considers lens distortion [3] [8] [12] [13], and one that neglects lens distortion [4] [6] [10] [11]. A typical linear technique that does not consider lens distortion is the one estimating the perspective transformation matrix $\mathbf{H}$ [6] [11]. The estimated $\mathbf{H}$ can be used *directly* for forward and backward 3D–2D projection. If necessary, given the estimated $\mathbf{H}$, the geometric camera parameters $\beta$ can be easily determined [5] [6] [10].

Faig's method [3] is a good representative for those considering lens distortion. For methods of this type, equations are established that relate the camera parameters to the 3D

object coordinates and 2D image coordinates of the calibration points. Nonlinear optimization techniques is then used to search for camera parameters with an objective to minimize residual errors of those equations. One disadvantage of this kind of method is that a good initial guess is required to start the nonlinear search.

A few years ago, Tsai proposed an efficient two–stage technique using the "radial alignment constraint" [12]. His method involves a direct solution for most of the calibration parameters and some iterative solution for the remaining parameters. Some drawbacks of Tsai's method are pointed out in [13]. Our experiences [7] also show that Tsai's method can be worse than the simple linear method of [11] if lens distortion is relatively small.

Recently, Weng shows some experimental results using a two–step method [13]. The first step involves a closed–form solution based on a distortion–free camera model, and the second step improves the camera parameters estimated in the first step by taking into account lens distortion. This method overcomes the initial guess problem in the nonlinear optimization, and is more accurate than Tsai's method according to our experiments.

We have also developed a fast and accurate technique for calibrating a camera, with lens distortion, by solving linear equations [8]. Instead of using nonlinear optimization techniques, the estimation of radial lens distortion coefficient is transformed into an eigenvalue problem of a $8 \times 8$ matrix. This method provides an efficient and accurate solution for calibrating a practical camera, and according to our experiment it is more accurate than Tsai's method.

However, considering lens distortion will not only complicate the camera calibration procedure, but also complicate the subsequent on–line processing such as feature–point correspondence (in stereo) and camera re–calibration (in the case of having a moving camera). Notice that epipolar line is no longer a straight line if lens distortion is taken into account. Moreover, when lens distortion is small, if the noise in the 2D feature extraction is relatively

large or the number of the calibration points is relatively small, the calibration results based on distortion camera model can be worse than those based on linear camera model. The question is then, "when should we consider lens distortion in camera calibration?" or "when does it worth all the troubles to consider lens distortion?" This work represents an effort toward the answer of the question.

## II. CAMERA MODEL

Consider the pinhole camera model with lens distortion, as shown in Fig. 1. Let $P$ be an object point in the 3D space, and $\mathbf{r}_O = (x_O \ y_O \ z_O)^t$ be its coordinates, in minimeters, with respect to a fixed object coordinate system (OCS). Let the camera coordinate system (CCS), also in minimeters, have its x–y plane parallel to the the image plane (such that x axis is parallel with the horizontal direction of the image, and y axis is parallel with the vertical one),

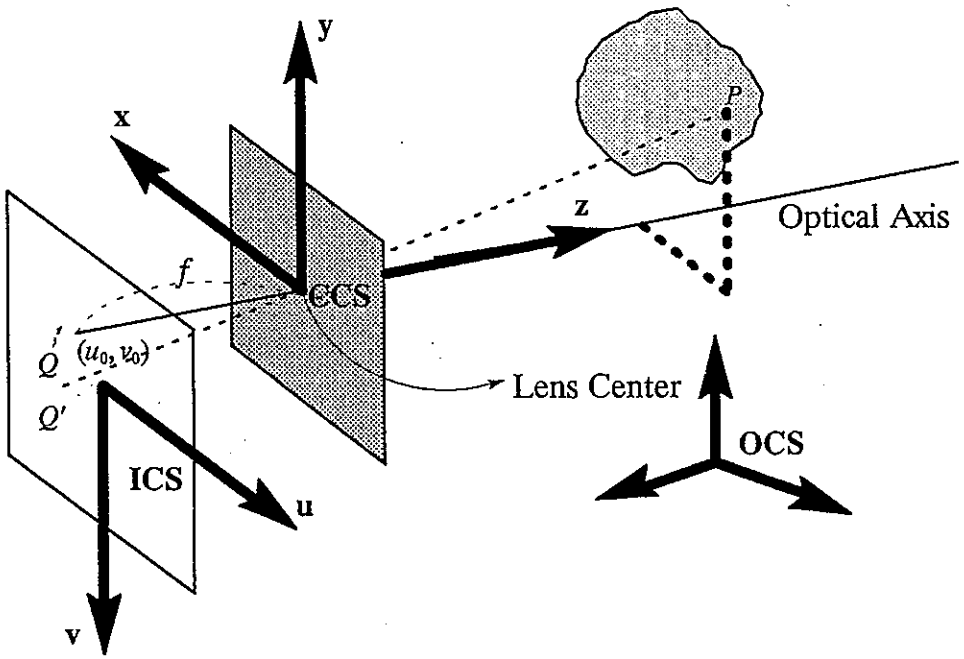


Fig. 1. Pinhole camera model with lens distortion, where $P$ is a 3D object point, $Q$ and $Q'$ are its undistorted and distorted image points, respectively.
OCS -- Object Coordinate System (3D)
CCS -- Camera Coordinate System (3D)
ICS -- computer Image Coordinate System (2D)

with its origin located at the lens center and its z axis aligned with the optical axis of the lens (see Fig. 1). Let $\mathbf{r}_C = (x_C \ y_C \ z_C)^t$ be the coordinates of the 3D point $P$ with respect to the CCS. Suppose there is no lens distortion, the corresponding image point of $P$ on the image plane would be $Q$ (see Fig. 1). However, due to the effect of lens distortion, the actual image point is $Q'$. Let $\mathbf{s}_I = (u_I \ v_I)^t$ denote the 2D image coordinates (in pixels), with respect to the computer image coordinate system (ICS), of the actual image point $Q'$, where the origin of ICS is located at *the center of the frame memory coordinate* (e.g. the origin of the ICS is right at (256, 256) for a 512 by 512 image).

As shown in Fig. 2, the 3D–2D transformation from $\mathbf{r}_O$ to $\mathbf{s}_I$ can be divided into the following four steps:

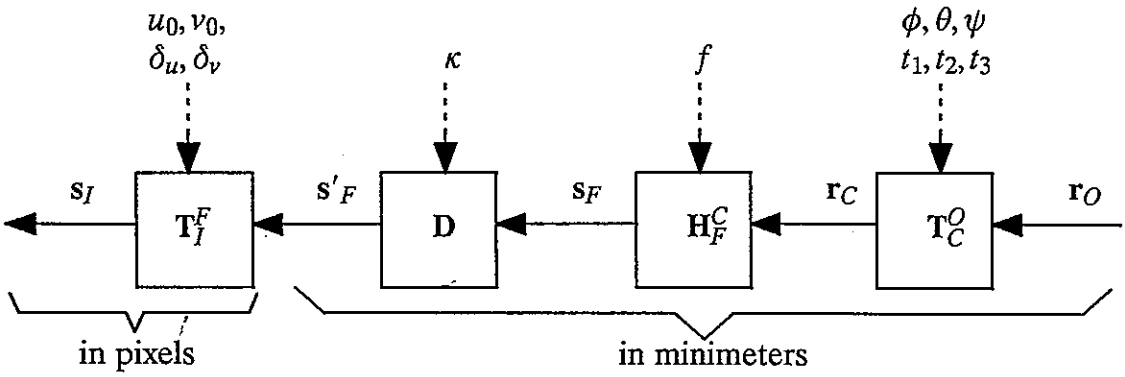


Fig. 2. Relation between different transformation matrices

**II.1 Translation and rotation from the OCS to the CCS**

The transformation from $\mathbf{r}_O$ to $\mathbf{r}_C$ can be expressed as

$$\tilde{\mathbf{r}}_C = \mathbf{T}_C^O \, \tilde{\mathbf{r}}_O \qquad \text{with} \qquad \mathbf{T}_C^O = \begin{bmatrix} \mathbf{R}_C^O & \mathbf{t}_C^O \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where tilde ( ˜ ) denotes homogeneous coordinates [2], $\mathbf{t}_C^Q = (t_1 \;\; t_2 \;\; t_3)^t$ is a translation vector, and $\mathbf{R}_C^Q$ is a 3x3 rotation matrix determined by the three Euler angles, $\phi$, $\theta$, $\psi$, rotating about the z, y, z axes sequentially.

## II.2 Perspective projection from a 3D object point in the CCS to a 2D image point on the image plane

Let $f$ be the "effective focal length", and let $\mathbf{s}_F = (u_F \;\; v_F)^t$ be the 2D coordinates (in minimeters) of the undistorted image point $Q$ lying on the image plane. Then, we have

$$u_F = f\frac{x_C}{z_C}, \quad v_F = f\frac{y_C}{z_C}. \tag{2.1}$$

Alternatively, we can express this perspective projection in the homogeneous coordinates as

$$\tilde{\mathbf{s}}_F = \mathbf{H}_F^C \, \tilde{\mathbf{r}}_C \qquad \text{with} \qquad \mathbf{H}_F^C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix}. \tag{2.2}$$

## II.3 Lens distortion from $Q$ to $Q'$

For practical reasons, we consider only the first term of the radial lens distortion, i.e.,

$$\mathbf{s}_F = (1 - \kappa \parallel \mathbf{s}'_F \parallel^2) \, \mathbf{s}'_F \tag{3}$$

where $\mathbf{s}'_F = (u'_F \;\; v'_F)^t$, is the coordinates of the distorted 2D image (in minimeters). In this paper, $\kappa$ has the unit of minimeter$^{-2}$.

## II.4 Scaling and translation of 2D image coordinates

The transformation from $\mathbf{s}'_F$ (in minimeters) to $\mathbf{s}_I$ (in pixels) involves (i) scaling from minimeter to pixels, and (ii) translation due to misalignment of the sensor array with the optical axis of the lens. Hence,

$$\bar{s}_I = \mathbf{T}_I^F \, \bar{s}'_F \qquad \text{with} \qquad \mathbf{T}_I^F = \begin{bmatrix} 1/\delta_u & 0 & u_0 \\ 0 & 1/\delta_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $\delta_u$ and $\delta_v$ are the horizontal and vertical pixel spacing (minimeter/pixel), $u_0$ and $v_0$ are the coordinates (in pixels) of the computer image coordinate system. For convenience, we will call $(u_0, v_0)$ the principle point, since it is the coordinates of the piercing point of the principle axis (as well as optical axis).

Using the above notations for camera parameters, $\beta = [t_1 \; t_2 \; t_3 \; \phi \; \theta \; \psi \; f \; \kappa \; \delta_u \; u_0 \; v_0]^t$. The vertical scaling factor $\delta_v$ is not included here because it is a known parameter when we use a solid state camera -- otherwise, only the ratios $f/\delta_u$ and $f/\delta_v$ can be determined. Combining (1), (2), (3) and (4), we have

$$(1 - \kappa\varrho^2)(u_I - u_0)\delta_u = f\frac{x_O r_1 + y_O r_2 + z_O r_3 + t_1}{x_O r_7 + y_O r_8 + z_O r_9 + t_3} \tag{5.1}$$

$$(1 - \kappa\varrho^2)(v_I - v_0)\delta_v = f\frac{x_O r_4 + y_O r_5 + z_O r_6 + t_2}{x_O r_7 + y_O r_8 + z_O r_9 + t_3} \tag{5.2}$$

where $\varrho = \sqrt{\delta_u^2(u_I - u_0)^2 + \delta_v^2(v_I - v_0)^2} = \| s'_F \|$.

Notice that, suppose there is no optical distortion (i.e., $\kappa = 0$ and $\mathbf{D}$ is an identity operator, see Fig. 2), the relationship between $\mathbf{r}_O$ and $\mathbf{s}_I$ can be expressed as a linear transformation by combining (1), (2) and (4):

$$\bar{s}_I = \mathbf{H} \, \bar{r}_O \qquad \text{i.e.,} \qquad \begin{bmatrix} u_I \cdot w \\ v_I \cdot w \\ w \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ h_5 & h_6 & h_7 & h_8 \\ h_9 & h_{10} & h_{11} & 1 \end{bmatrix} \begin{bmatrix} x_O \\ y_O \\ z_O \\ 1 \end{bmatrix} \tag{6}$$

where $\mathbf{H} = \mathbf{T}_I^F \, \mathbf{H}_F^C \, \mathbf{T}_C^O$ .

Hereafter, for simplicity, we will use $u$, $v$, $x$, $y$, $z$ to denote $u_I$, $v_I$, $x_O$, $y_O$, $z_O$, respectively.

## III. A LINEAR METHOD FOR CAMERA CALIBRATION

Given a set of 3D calibration points and their corresponding 2D image coordinates, the problem is to estimate $\beta$, the parameters of our camera model. Instead of estimating $\beta$ directly, we first estimate the composite parameters $\mathbf{h}$ (as described following equation(8)), then the composite parameters can be decomposed into $\beta$ by methods described in [5] [6] [10].

From equation (6), the derivation of a linear method for camera calibration is quite simple (refer to [5] [6] [10]). However, to observe the effects of lens distortion on camera calibration, it is necessary to derive the linear method in a different way.

From (5.1) and (5.2) it can be shown that (refer to [9])

$$\mathbf{Ap} + \mathbf{Bq} + \kappa\mathbf{Cq} = 0\,, \tag{7}$$

where
$$\mathbf{A} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \quad \mathbf{B} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_j x_j & -u_j y_j & -u_j z_j & -u_j \\ -v_j x_j & -v_j y_j & -v_j z_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

$$\mathbf{C} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ (u_j - u_0)\varrho_j^2 x_j & (u_j - u_0)\varrho_j^2 y_j & (u_j - u_0)\varrho_j^2 z_j & (u_j - u_0)\varrho_j^2 \\ (v_j - v_0)\varrho_j^2 x_j & (v_j - v_0)\varrho_j^2 y_j & (v_j - v_0)\varrho_j^2 z_j & (v_j - v_0)\varrho_j^2 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

$$\mathbf{P}_1 \equiv \begin{bmatrix} r_1 f/\delta_u + r_7 u_0 \\ r_2 f/\delta_u + r_8 u_0 \\ r_3 f/\delta_u + r_9 u_0 \\ t_1 f/\delta_u + t_3 u_0 \end{bmatrix}, \quad \mathbf{P}_2 \equiv \begin{bmatrix} r_4 f/\delta_v + r_7 v_0 \\ r_5 f/\delta_v + r_8 v_0 \\ r_6 f/\delta_v + r_9 v_0 \\ t_2 f/\delta_v + t_3 v_0 \end{bmatrix}, \quad \mathbf{P}_3 \equiv \begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix},$$

and
$$\mathbf{p} \equiv \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix} / t_3 = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_8 \end{bmatrix}, \quad \mathbf{q} \equiv \mathbf{P}_3 / t_3 = \begin{bmatrix} r_7/t_3 \\ r_8/t_3 \\ r_9/t_3 \\ 1 \end{bmatrix} = \begin{bmatrix} h_9 \\ h_{10} \\ h_{11} \\ 1 \end{bmatrix}.$$

Suppose we have a distortionless lens, i.e., $\kappa = 0$, then equation (7) can be simplified to

$$\mathbf{Ap} + \mathbf{Bq} = [\mathbf{A}\ \ \mathbf{B}] \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} = [\mathbf{A}\ \ \mathbf{B}' \vdots -\mathbf{b}] \begin{bmatrix} \mathbf{h} \\ 1 \end{bmatrix} = \mathbf{e} \approx 0, \tag{8}$$

where $\mathbf{B}'$ is the matrix obtained by removing the last column of $\mathbf{B}$, $\mathbf{b} \equiv [\ .. \ u_j \ v_j \ .. \ ]^t$ (i.e., $[\mathbf{B}' \vdots -\mathbf{b}] = \mathbf{B}$) and $\mathbf{h} = [h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8 h_9 h_{11}]^t$. Notice that the small perturbation, $\mathbf{e}$, in equation (8) is due to the measurement error of the 3D and 2D coordinates of the calibration points.

Hence, the parameters to be estimated, $\mathbf{h}$, can be computed by minimizing the following error function, $\| \mathbf{e} \|^2$, with respect to $\mathbf{h}$:

$$\| \mathbf{e} \|^2 = \| \Lambda\ \mathbf{h} - \mathbf{b} \|^2, \tag{9}$$

where $\Lambda \equiv [\mathbf{A}\ \ \mathbf{B}']$.

The optimal solution of (9) is well known to be

$$\hat{\mathbf{h}} = (\Lambda^t \Lambda)^{-1} \Lambda^t\ \mathbf{b}, \tag{10}$$

if there are more than *six noncoplanar calibration points*. The estimated composite parameters can be further decomposed into $\beta$, when necessary [6].

## IV. ACCURACY ASSESSMENT

In this section we will derive an approximate error bound for the linear calibration method shown in section III, which does not consider lens distortion. The effects of both the measurement noise and the modeling error (the negligence of lens distortion) are considered.

The error bound is based on the following assumptions:

(a.1). The 3D positions of the calibration points are known exactly. In practice, the 3D position of a calibration point is more easy to be located precisely comparing to its 2D image coordinates. Furthermore, the 3D position error can always be transformed to an equivalent 2D measuring error.

(a.2). The only source of measurement noise is the error in estimating the image coordinates of the calibration points, i.e., the 2D observation noise (in pixels). In both horizontal and vertical directions, we assume the 2D observation noise have the identically independent Gaussian distribution with zero mean and the variance, $\sigma^2$.

(a.3). The depth components of both calibration and test points, $z_C$, can be approximately replaced by a constant (i.e. the depth of field is small relative to object distance). This assumption holds in most computer vision applications, since the depth of field for a practical camera is usually limited to a small range comparing to the object distance.

**IV.1 Definition of error measure**

To evaluate the accuracy of the camera calibration for 3D vision application, it is necessary to define certain kinds of error measure. The measures adopted in this paper are (refer to Fig. 3):

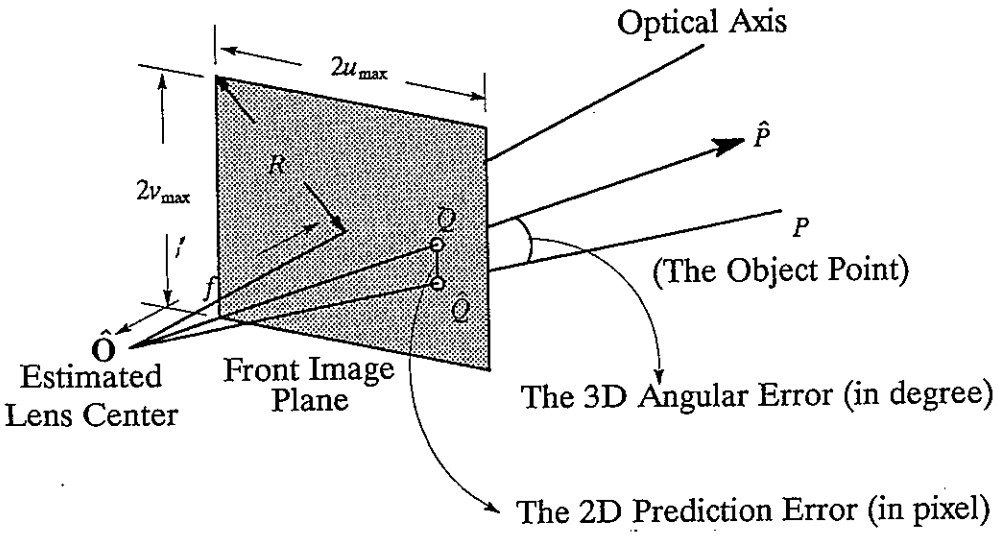


Fig. 3. The error measures adopted in this paper.

(i). the *3D angular error*, i.e, the angle (in degree) $\measuredangle P\hat{O}\hat{P}$ where $P$ is the 3D test point, $\hat{O}$ is the estimated lens center, and $\hat{O}\hat{P}$ is the 3D ray back projected from the observed 2D image of $P$.

(ii). the *2D prediction error*, i.e, the image distance (in pixels) between the $Q$ and $\hat{Q}$, where $Q$ is the true 2D image coordinates of the test point $P$, and $\hat{Q}$ is the predicated image coordinates of the 3D test point $P$ using the estimated parameters.

There is a relationship between the 2D prediction error and the 3D angular error (see Fig. 3), which can be approximated by

$$3D \ \ angular \ \ error \approx 2D \ \ prediction \ \ error * \frac{\delta_a}{d_a}, \tag{11}$$

where $d_a$ denotes the average distance from the image point to the estimated lens center, and $\delta_a$ stands for the average pixel spacing (see Appendix I).

In the following, we are going to give the intuition of the error function (9). Hereafter, for convenience, we will use $(\hat{u}, \hat{v})$ to denote the predicted 2D image coordinates of the 3D calibration point of $(x, y, z)$, i.e.,

$$\hat{u} \equiv \frac{\hat{h}_1 x + \hat{h}_2 y + \hat{h}_3 z + \hat{h}_4}{\hat{h}_9 x + \hat{h}_{10} y + \hat{h}_{11} z + 1}, \tag{12.1}$$

and

$$\hat{v} \equiv \frac{\hat{h}_5 x + \hat{h}_6 y + \hat{h}_7 z + \hat{h}_8}{\hat{h}_9 x + \hat{h}_{10} y + \hat{h}_{11} z + 1}. \tag{12.2}$$

Notice that the error function (9) is equivalent to the following equation (refer to [9])

$$\| \mathbf{e} \|^2 = \| \Lambda \ \mathbf{h} - \mathbf{b} \|^2 = \sum_{j=1}^{N_{calib}} \left[ (u_j - \hat{u}_j)^2 \left( \frac{\hat{z}_{Cj}}{\hat{t}_3} \right)^2 + (v_j - \hat{v}_j)^2 \left( \frac{\hat{z}_{Cj}}{\hat{t}_3} \right)^2 \right], \tag{13}$$

where $\frac{\hat{z}_{Cj}}{\hat{t}_3} = \hat{h}_9 x_j + \hat{h}_{10} y_j + \hat{h}_{11} z_j + 1$.

In equation (13), the 2D prediction error of each calibration point is weighted by the factor $z_C/t_3$, which means that the linear calibration method tends to minimize the 2D prediction error of those points far away from the camera. But, from assumption (a.3) and equation (13), we have

$$\| \mathbf{e} \|^2 \approx \left\{ \sum_{j=1}^{N_{calib}} \left[ (u_j - \hat{u}_j)^2 + (v_j - \hat{v}_j)^2 \right] \right\} \times \left( \frac{z_C}{t_3} \right)^2, \tag{14}$$

which amounts to say that (10) is the *optimal solution* that minimizes the 2D prediction error.

**IV.2 The 2D prediction error as a function of the number of calibration points and 2D observation noise.**

Consider the ideal case that both 2D and 3D coordinates are noise free, equation (8) can be written as

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix} + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_{truej}x_j & -u_{truej}y_j & -u_{truej}z_j & -u_{truej} \\ -v_{turej}x_j & -v_{truej}y_j & -v_{truej}z_j & -v_{truej} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \mathbf{P}_3 = 0. \tag{15}$$

Now suppose the observation noise along u–axis (v–axis) is $n_u$ ($n_v$), i.e.,

$$u = u_{true} + n_u \text{ and } v = v_{true} + n_v, \tag{16}$$

where $(u_{true}, v_{true})$ is the true image coordinates (noise free) and $(u, v)$ is the measured one (noisy). Substituting $u_{true}$ and $v_{true}$ in (16) into (15), we have

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix} + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_jx_j & -u_jy_j & -u_jz_j & -u_j \\ -v_jx_j & -v_jy_j & -v_jz_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \mathbf{P}_3$$

$$+ \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ n_{uj}x_j & n_{uj}y_j & n_{uj}z_j & n_{uj} \\ n_{vj}x_j & n_{vj}y_j & n_{vj}z_j & n_{vj} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \mathbf{P}_3 = 0. \tag{17}$$

From the definition of $\mathbf{P}_3$, we have [9]

$$[x_j \;\; y_j \;\; z_j \;\; 1]\; \mathbf{P}_3 = z_{Cj}\,, \tag{18}$$

where $z_{Cj}$ is the z–component of the coordinates of the $j$th calibration point in the camera coordinate system. Substituting (18) into (17) and dividing (17) by $t_3$, it follows that

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \mathbf{p} + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_j x_j & -u_j y_j & -u_j z_j & -u_j \\ -v_j x_j & -v_j y_j & -v_j z_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \mathbf{q} + \begin{bmatrix} \vdots \\ n_{uj} z_{Cj}/t_3 \\ n_{vj} z_{Cj}/t_3 \\ \vdots \end{bmatrix} = 0. \tag{19}$$

Comparing equations (8) and (19), we have

$$\mathbf{e} = -\begin{bmatrix} \vdots \\ n_{uj} z_{Cj}/t_3 \\ n_{vj} z_{Cj}/t_3 \\ \vdots \end{bmatrix} \approx -\begin{bmatrix} \vdots \\ n_{uj} \\ n_{vj} \\ \vdots \end{bmatrix} z_C/t_3\,. \tag{20}$$

Using equations (8) and (9), we have

$$\Lambda\, \mathbf{h} = \mathbf{b} + \mathbf{e} \equiv \mathbf{b}'\,. \tag{21}$$

In practice, both the calibration and the test points are selected from the same working volume. Thus the 2D prediction error calculated by using the test points can be approximated by the one using the noise–free version of the calibration points. Therefore, the following work is to find the 2D prediction error calculated by using the noise free calibration points. Let us denote the expectation of the average 2D prediction error as $\epsilon_n$, then we have

$$\epsilon_n^2 \equiv \frac{1}{N_{calib}} E \sum_{j=0}^{N_{calib}} \left[ \left(u_{truej} - \hat{u}_j\right)^2 + \left(v_{truej} - \hat{v}_j\right)^2 \right]. \tag{22}$$

By assumption (a.3), we have

$$N_{calib} \times \left(\frac{z_C}{t_3}\right)^2 \times \epsilon_n^2 \approx E \sum_{j=0}^{N_{calib}} \left[ \left(u_{truej} - \hat{u}_j\right)^2 \left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right)^2 + \left(v_{truej} - \hat{v}_j\right)^2 \left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right)^2 \right]. \tag{23}$$

From equations (21) and (13), equation (23) can be further simplified as

$$N_{calib} \times \left(\frac{z_C}{t_3}\right)^2 \times \epsilon_n^2 \;=\; E \,\| \mathbf{b}' - \Lambda \hat{\mathbf{h}} \,\|^2 . \tag{24}$$

Since equation (21) is derived from a noise–free equality (17), it is obvious that the solution of (21) obtained by pseudo inverse has zero residual error, i.e.

$$\mathbf{b}' - \Lambda \mathbf{h} \;=\; \mathbf{b}' - \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\,\mathbf{b}' \;=\; (\,\mathbf{b} + \mathbf{e}\,) \;-\; \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\,(\,\mathbf{b} + \mathbf{e}\,) \;=\; \mathbf{0}. \tag{25}$$

Thus we have

$$\| \mathbf{b}' - \Lambda \hat{\mathbf{h}} \|^2 = \| \mathbf{b} + \mathbf{e} - \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\mathbf{b} \|^2 = \| \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\mathbf{e} \|^2 . \tag{26}$$

The expectation of equation (26) is

$$\begin{aligned}
& E \,\| \mathbf{b}' - \Lambda \hat{\mathbf{h}} \|^2 \\
&= E\Big\{ \| \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\mathbf{e} \|^2 \Big\} = E\Big\{ \mathbf{e}^t\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\mathbf{e} \Big\} \\
&= E\Big\{ \mathbf{e}^t\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\mathbf{e} \Big\} \;=\; trace\Big[ \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t \Big] \sigma^2 z_C^2/t_3^2 \\
&= trace\Big[ (\Lambda^t\Lambda)^{-1}\Lambda^t\Lambda \Big] \sigma^2 z_C^2/t_3^2 = 11\sigma^2 z_C^2/t_3^2 .
\end{aligned} \tag{27}$$

Taking average of the above equation over the $N_{calib}$ points and dividing it with the constant $z_C^2/t_3^2$, we have the *expectation* of the average square 2D prediction error

$$\epsilon_n^2 \equiv \frac{11\ \sigma^2}{N_{calib}} \qquad \text{(in pixels)} \tag{28}$$

**IV.3 The modeling error**

Suppose the 2D–3D pairs of the calibration points are noiseless, but the lens we used has certain amount of lens distortion, i.e., $\kappa \neq 0$. And if a linear calibration method which dose not consider lens distortion is used, then the deduced 2D prediction error is called the modeling error. Usually, the 2D prediction error has two sources, one is the effect of the

measuring noise, which has been discussed in the previous sub–section. Another is due to the improper modeling of the camera, which will be dealt with in this sub–section.

Knowing that the linear calibration method will minimize the 2D prediction error, see equation (14), subject to the distortion free camera model and providing that the assumption (a.3) is true. The idea of the following work is to find an approximate solution that is close to the optimal solution and convenient for us to compute its 2D prediction error. The 2D prediction error of the approximate solution is then used as the upper bound, since the optimal solution will always have a smaller error. *The more approximations included in the derivation of the approximate solution the more conservative the bound is*, because of that the approximate solution deviates from the optimal solution more. The derivation of the upper bound of the modeling error is described separately in the following steps.

**IV.3.1. Find an approximate relation between the estimated composite parameters and the true one.**

Rewrite (7) as following

$$\Lambda\mathbf{h} = \mathbf{b} - \kappa \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ (u_j - u_0)\varrho_j^2 x_j & (u_j - u_0)\varrho_j^2 y_j & (u_j - u_0)\varrho_j^2 z_j & (u_j - u_0)\varrho_j^2 \\ (v_j - v_0)\varrho_j^2 x_j & (v_j - v_0)\varrho_j^2 y_j & (v_j - v_0)\varrho_j^2 z_j & (v_j - v_0)\varrho_j^2 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \mathbf{P}_3 / t_3 \,. \tag{29}$$

Since $z_{Cj} = [x_j \;\; y_j \;\; z_j \;\; 1]\mathbf{P}_3$, $z_{C1} \approx z_{C2} \approx \ldots \approx z_{CN} \approx z_C$, and in general, the principle point, $(u_0, v_0)$, is negligible comparing to the image points, $(u_j, v_j)$, we have

$$\Lambda\mathbf{h} \approx \mathbf{b} - \kappa \frac{z_C}{t_3} \begin{bmatrix} \vdots \\ (u_j - u_0)\varrho_j^2 \\ (v_j - v_0)\varrho_j^2 \\ \vdots \end{bmatrix} \approx \mathbf{b} - \kappa \frac{z_C}{t_3} \begin{bmatrix} \vdots \\ (u_j)\varrho_j^2 \\ (v_j)\varrho_j^2 \\ \vdots \end{bmatrix}. \tag{30}$$

Now suppose that $\varrho_j^2$ can be approximated to a constant, i.e., $\varrho_j^2 \approx M$, for all $j$, where $M$ is a constant to be determined later. Substituting the above approximations to (30), yields

$$\Lambda\ \mathbf{h} \approx (1 - \kappa M z_C / t_3)\mathbf{b}. \tag{31}$$

From (31), we have

$$\mathbf{h}_{true} \approx (1 - \kappa M z_C / t_3)\ \left(\Lambda^t \Lambda\right)^{-1} \Lambda^t\ \mathbf{b}\ =\ (1 - \kappa M z_C / t_3)\ \hat{\mathbf{h}}\,, \tag{32}$$

where $\mathbf{h}_{true}$ is an exact solution of equation (29).

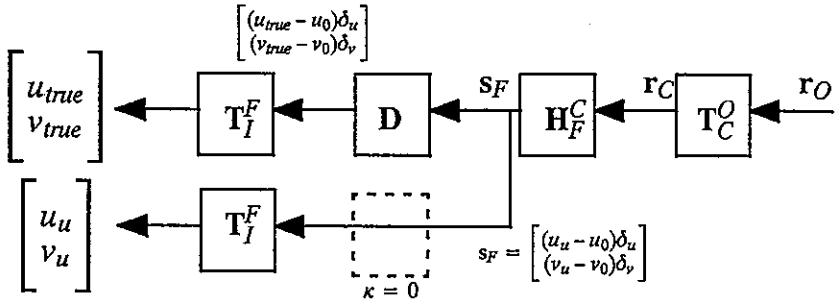


Fig. 4(a). This figure shows the relation between the true image point $[u_{true}, v_{true}]^t$ and its undistorted image point, $[u_{true}, v_{true}]^t$, when the true parameters are known.

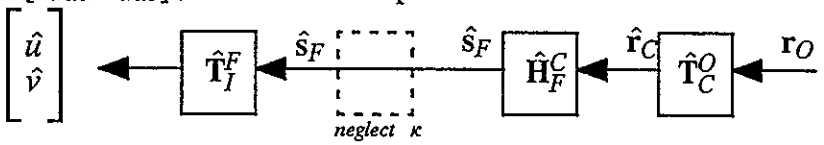


Fig. 4.(b). The 2D predicted image $[\hat{u}\ \hat{v}]^t$ is computed using the camera parameters estimated using linear calibration method without considering lens distortions.

**IV.3.2. Find the relation of the true image point $(u_{true}, v_{true})$ and its undistorted image point $(u_u, v_u)$. (refer to Fig. 4.(a))**

Using the estimated parameters $\hat{\mathbf{h}}$ the predicted image point $(\hat{u},\ \hat{v}\ )$ (in pixels) is defined in equation (12). Also, if the true parameters $\mathbf{h}_{true}$ were used, then the the undistorted image point $(u_u,\ v_u\ )$ (in pixel) will be

$$u_u = \frac{h_1 x + h_2 y + h_3 z + h_4}{h_9 x + h_{10} y + h_{11} z + 1}, \tag{33.1}$$

$$v_u = \frac{h_5 x + h_6 y + h_7 z + h_8}{h_9 x + h_{10} y + h_{11} z + 1}, \tag{33.2}$$

where the subscript 'u', denotes that the coordinate is undistorted, i.e., to obtain the correct coordinates, one further step is necessary to compensate the effects of the lens distortion. From equation (4), we have

$$\mathbf{s}'_F = \begin{bmatrix} (u-u_0)\delta_u \\ (v-v_0)\delta_v \end{bmatrix} \text{ and } \mathbf{s}_F = \begin{bmatrix} (u_u-u_0)\delta_u \\ (v_u-v_0)\delta_v \end{bmatrix}, \tag{34}$$

In practice, $\kappa\varrho^2 \ll 1$ (e.g. $\kappa = 0.00035$ minimeter$^{-2}$, $\varrho^2_{\max} = 25.80$ minimeters$^2$, and $\kappa\varrho^2_{\max} = 0.01$). Therefore from equation (3), it follows that

$$\begin{bmatrix} (u_{true}-u_0) \\ (v_{true}-v_0) \end{bmatrix} = 1/(1-\kappa\varrho^2)\begin{bmatrix} (u_u-u_0) \\ (v_u-v_0) \end{bmatrix} \approx (1+\kappa\varrho^2)\begin{bmatrix} (u_u-u_0) \\ (v_u-v_0) \end{bmatrix}. \tag{35}$$

**IV.3.3. Find the relation of the predicted 2D image point $(\hat{u},\hat{v})$ and its true undistorted image point $(u_u, v_u)$. (refer to Figs. 4.(a) and (b))**

By equation (12), (32) and (33), it can be shown that (refer to [9]).

$$\begin{bmatrix} \hat{u}-u_0 \\ \hat{v}-v_0 \end{bmatrix} \approx (1+\kappa M)\begin{bmatrix} u_u-u_0 \\ v_u-v_0 \end{bmatrix} \tag{36}$$

**IV.3.4. Find the error of the predicted 2D image point $(\hat{u},\hat{v})$ with respect to its true image point $(u_{true}, v_{true})$. (refer to Figs. 4.(a) and (b))**

By (35) and (36), the error of the predicted 2D image point, $(\hat{u},\hat{v})$, with respect to the corresponding true image point, $(u_{true}, v_{true})$ is:

$$\begin{bmatrix} (u_{true}-\hat{u}) \\ (v_{true}-\hat{v}) \end{bmatrix} = \begin{bmatrix} (u_{true}-u_0)-(\hat{u}-u_0) \\ (v_{true}-u_0)-(\hat{v}-v_0) \end{bmatrix} \approx \begin{bmatrix} (u_u-u_0)(1+\kappa\varrho^2)-(u_u-u_0)(1+\kappa M) \\ (v_u-v_0)(1+\kappa\varrho^2)-(v_u-v_0)(1+\kappa M) \end{bmatrix}$$

$$= \begin{bmatrix} \kappa(u_u-u_0)(\varrho^2-M) \\ \kappa(v_u-v_0)(\varrho^2-M) \end{bmatrix}. \tag{37}$$

For convenience, we would like to calculate the mean square 2D prediction error by multiplying both the errors in two directions by their scale factors, $\delta_u, \delta_v$, respectively. Let $U$ and $V$ denote $(u_u - u_0)$ and $(v_u - v_0)$, respectively. As explained before, $\kappa\varrho^2 \ll 1$. Hence, from (37), the 2D square error becomes

$$\left\| \begin{bmatrix} \delta_u(u_{true} - \hat{u}) \\ \delta_v(v_{true} - \hat{v}) \end{bmatrix} \right\|^2 \approx \left\| \begin{bmatrix} \kappa\ \delta_u\ U(\varrho^2 - M) \\ \kappa\ \delta_u\ V(\varrho^2 - M) \end{bmatrix} \right\|^2 \approx \kappa^2\ \hat{\varrho}^2\ (\varrho^2 - M), \tag{38}$$

where $\hat{\varrho}^2 \equiv (U\delta_u)^2 + (V\delta_v)^2$ and $\varrho^2 = ((u-u_0)\delta_u)^2 + ((v-v_0)\delta_v)^2 \approx \hat{\varrho}^2$.

Now, the mean square 2D error is calculated using the following equations

$$\begin{aligned} \epsilon_M^2 &= \frac{1}{u_{\max} v_{\max}} \int_0^{u_{\max}} \int_0^{v_{\max}} \kappa^2(\varrho^2 - M)^2((U\delta_u)^2 + (V\delta_v)^2) dU dV, \\ &\approx \frac{1}{u_{\max} v_{\max}} \int_0^{u_{\max}} \int_0^{v_{\max}} \kappa^2(\hat{\varrho}^2 - M)^2 \hat{\varrho}^2 dU dV, \end{aligned} \tag{39}$$

where $\epsilon_M^2$ is the mean square of the modeling error, and $u_{max}$, $v_{max}$ are the length, in pixels, of half the maximal size (in pixels) of the image in either direction (see Fig. 3). If we integrate the error on the disk whose radius, $R = \sqrt{(\delta_u u_{\max})^2 + (\delta_v v_{\max})^2}$, equals to half the diagonal size of the image sensor, then we will have

$$\begin{aligned} \epsilon_M^2 &\approx \frac{1}{\pi\ R^2} \int_0^R \int_0^{2\pi} \kappa^2(\hat{\varrho}^2 - M)^2 \hat{\varrho}^2 d\theta\ d\varrho \\ &= \kappa^2 [\frac{R^6}{8} - \frac{2R^4 M}{6} + \frac{R^2 M^2}{4}] \quad \text{(in minimeters)} \end{aligned} \tag{40}$$

Minimize (40) subject to $M$, we have

$$M = 2R^2/3, \text{ and } \epsilon_M^2 = \kappa^2 R^6/36. \tag{41}$$

Recall that we claim that $\varrho_j^2$ can be approximated to a constant, which yields equation (32), but in practice, this is usually not a good approximation. Besides, since $\hat{\mathbf{h}}$ is the optimal

solution that minimize $\epsilon_M^2$, and $(1 + \kappa M z_C/t_3)\mathbf{h}_{true}$ (an approximation) is used to replace $\hat{\mathbf{h}}$ (the optimal solution) to calculate the 2D error, so that the obtained results is the upper bound of the 2D error (in minimeters).

With the average pixel spacing (see Appendix I), the 2D error bound can be represented in pixel, which yields

$$\epsilon_M^2 = \kappa^2 \frac{R^6}{36\delta_a^2} \quad \text{(in pixels)} \tag{42}$$

Many techniques can be used to determine the value of $\kappa$, but we recommend to use the method we proposed in [8], since least efforts is needed to adapt the method described in section III to estimate $\kappa$.

**IV.4 The bound of total 2D prediction error**

Assume that the interaction between measurement noise and modeling error is small. Then, we have the approximate total mean square 2D prediction error combining (28) and (42):

$$\epsilon_{Bound}^2 \approx \epsilon_M^2 + \epsilon_n^2 . \tag{43}$$

Notice that the second term, $\epsilon_n^2$, of equation (43) is an expectation value, which means that the violation of the approximate upper bound, $\epsilon_{Bound}$, is possible.

## V. EXPERIMENTAL RESULTS

In this section, we will show some experimental results obtained by both computer simulations and real experiments. In the simulations, we assume the 3D positions of the calibration points are known exactly, and the only source of measurement noise is the error in estimating the image coordinates of the calibration points, i.e., the 2D observation noise. The reason for doing so in the simulation is because, for our applications, the 3D measurement

noise is easier to be controlled such that it has much smaller effect than the 2D observation noise has. Let $\sigma$ denote the standard deviation of the 2D observation noise. Unless specified explicitly, the following parameters are used in the simulations (most of these parameters are obtained from a real experiment using nonlinear calibration method [13]). The images are of 480x512 pixels. The synthetic camera is assumed to have the effective focal length of $f$ = 25.8547 minimeters and the pixel size of $\delta_u$ = 0.01566 minimeters and $\delta_v$ = 0.013 minimeters in horizontal and vertical directions, respectively. The radial lens distortion coefficient is 0.00035 minimeter$^{-2}$. The extrinsic camera parameters include three Euler angles, 45.22, 0.95 and 45.52 all in degrees, rotating about z–, y–, z– axes successively, and the transition vector (138.82, 136.81, 1811.11) minimeters. The calibration and test points are selected from a volume having the depth (in the direction of the optical axis) of 500 minimeters.

The first experiment observes the effects of both the 2D observation noise and the lens distortion in camera calibration using distortionless model. Each simulated data point shown in Fig. 5 is the average of ten random trials, while the number of calibration points are set to

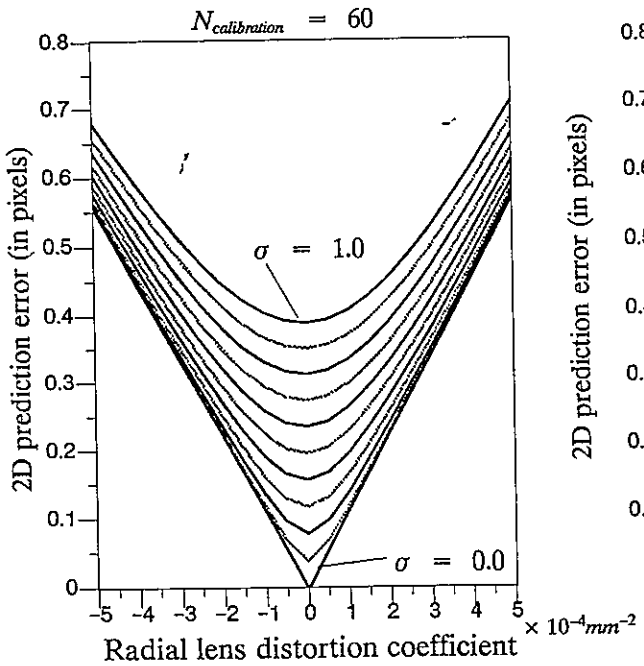


Fig. 5. The simulated 2D prediction error

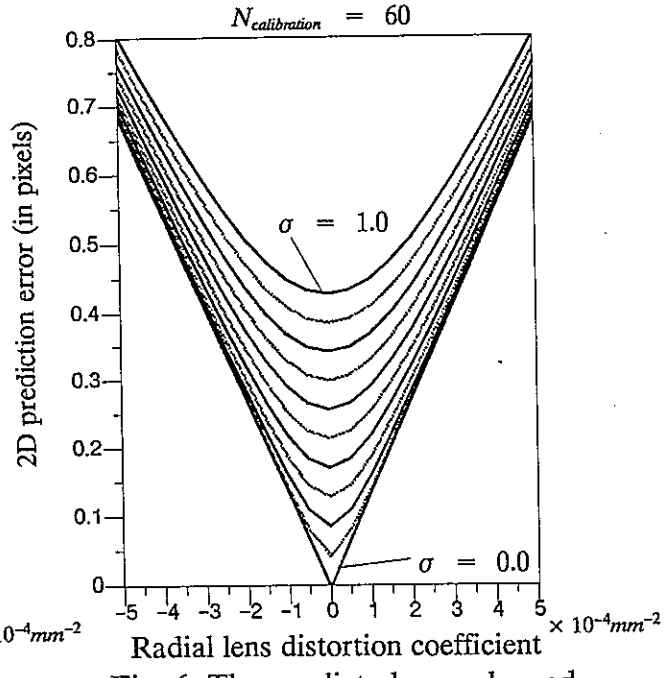


Fig. 6. The predicted error bound

60 points. As shown by the V–shape curves in Fig. 5, the 2D prediction error is proportional to the amount of lens distortion, i.e. $\epsilon_{Bound} \propto \kappa$, as we expected. Also, it increases as the 2D observation noise or $\sigma$ increases, i.e. $\epsilon_{Bound} \propto \sigma$. Each curve shown in Fig. 5, from bottom to top is obtained by using the 2D observation noise having the standard deviation, $\sigma = 0.0$, 0.1, 0.2, ..., 1.0 pixels, respectively. Fig. 6 shows the error bound obtained by using equation (43). Although the basic assumption (a.3) does not hold (z–components can vary in the range of 1300 to 1800 minimeters), our error bound still predicts the actual 2D prediction error quite precisely.

This bound is tested further by the next experiment. Here, four of the intrinsic parameters are generated randomly (see Table 1). The calibration and test points are generated from 20 planes, which are equally spaced with $Z_{inc}$ minimeters. Hence, we are using a working volume having the depth of $20 \times Z_{inc}$ minimeters. On each plane, we generate $N_p$ random points for calibration, which yields totally $N_{calib} = 20 \times N_p$ calibration points. The reason we set up such a configuration is to simulate the real equipment we have. Both $Z_{inc}$ and $N_p$

| Parameters | Interval of the uniform distribution |
|---|---|
| focal length, $f$ | 12.5 mm ~ 75 mm |
| principle point, $u_0$ | –20 pixels ~ +20 pixles |
| principle point, $v_0$ | –20 pixels ~ +20 pixles |
| lens distortion, $\kappa$ | $-0.0005\text{mm}^{-2}$ ~ $+0.0005\text{mm}^{-2}$ |
| 2D noise, $\sigma$ | 0.0 pixel ~ 1.0 pixel |
| Distance between successive plane, $Z_{inc}$ | 1 mm ~ 25 mm |
| # of calibration points on each plane, $N_p$ | 1 point ~ 10 points* |

* integer random number

Table 1. the interval of parameters tested in experiment 2.

in this experiment are also randomly selected (as shown in Table 1), but the number of test points is fixed to 200.

Totally, ten thousand trials were simulated. For each random trial, the computed 2D prediction error is normalized by its theoretic bound. Fig. 7 shows the histogram of the normalized error which shows that, in most trials the 2D prediction error is close to and less than

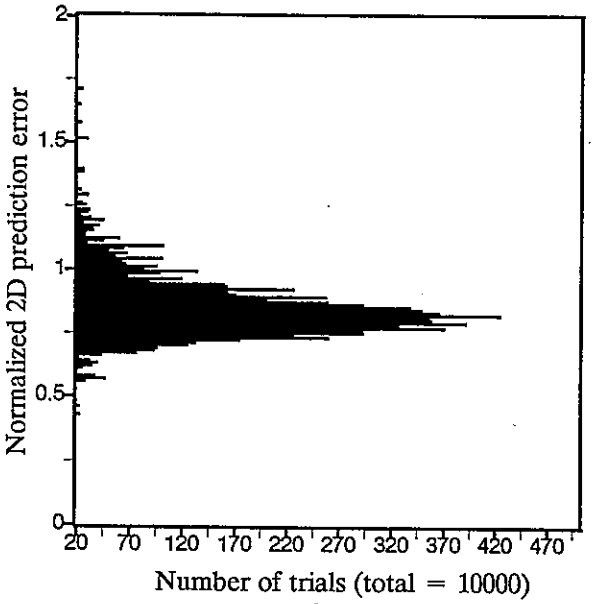

Fig. 7.

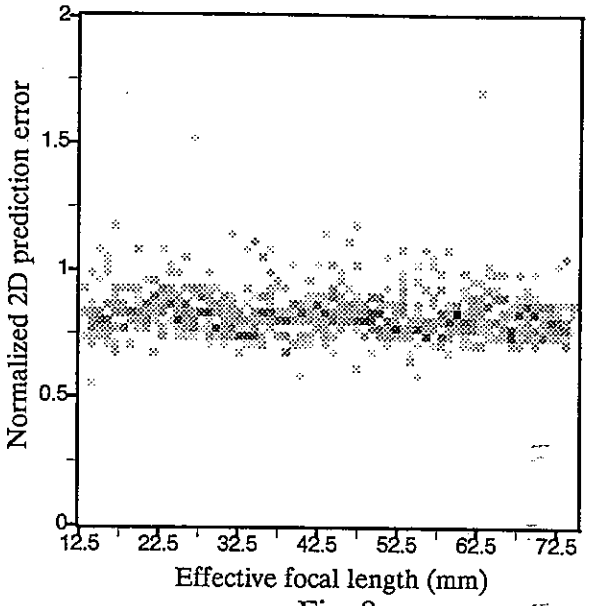

Fig. 8.

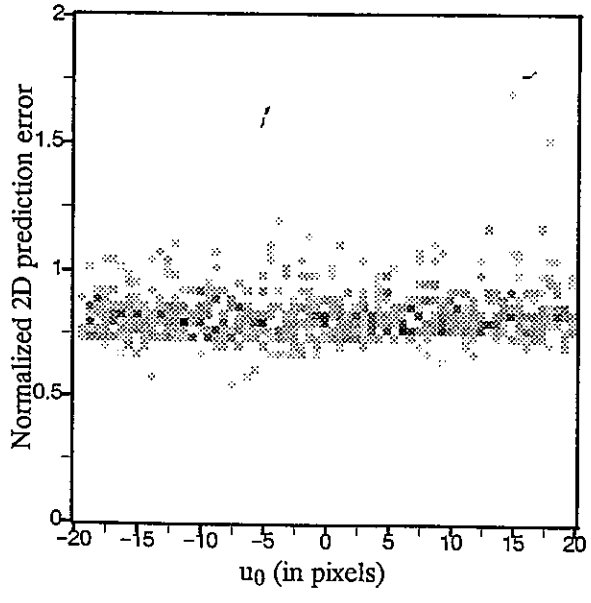

Fig. 9.

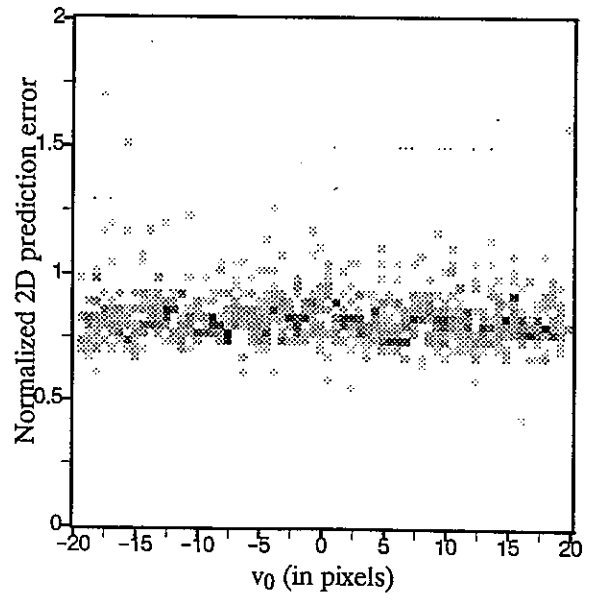

Fig. 10.

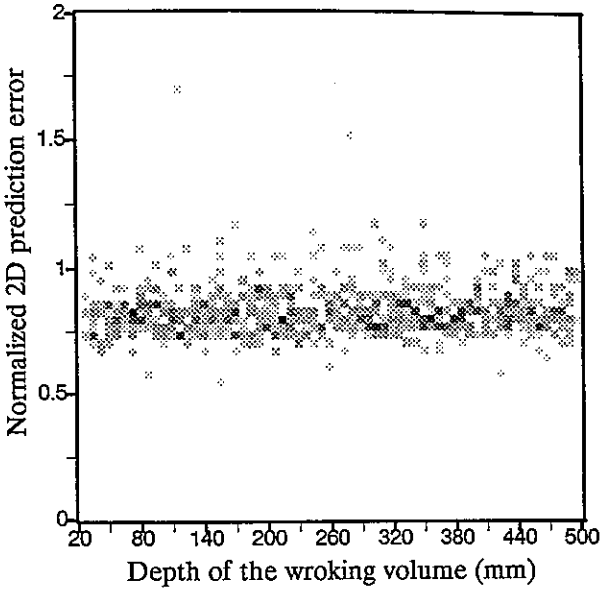


Fig. 11.

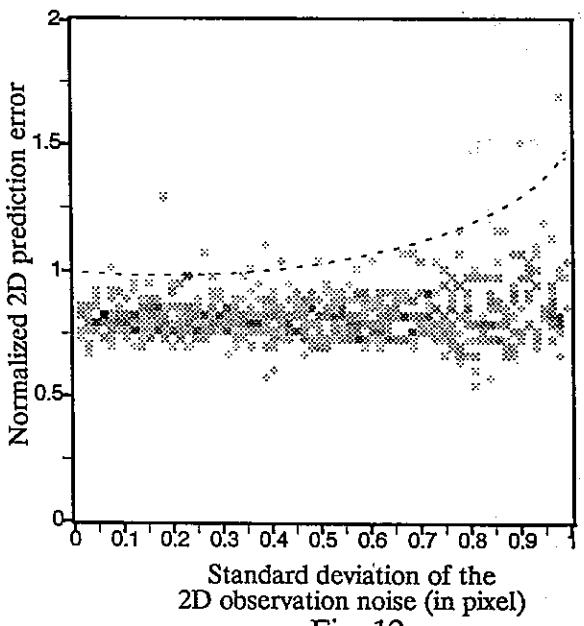


Fig. 12.

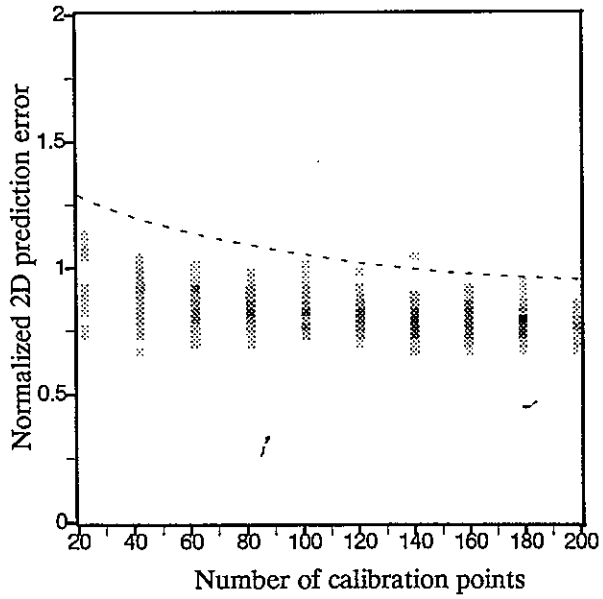


Fig. 13.

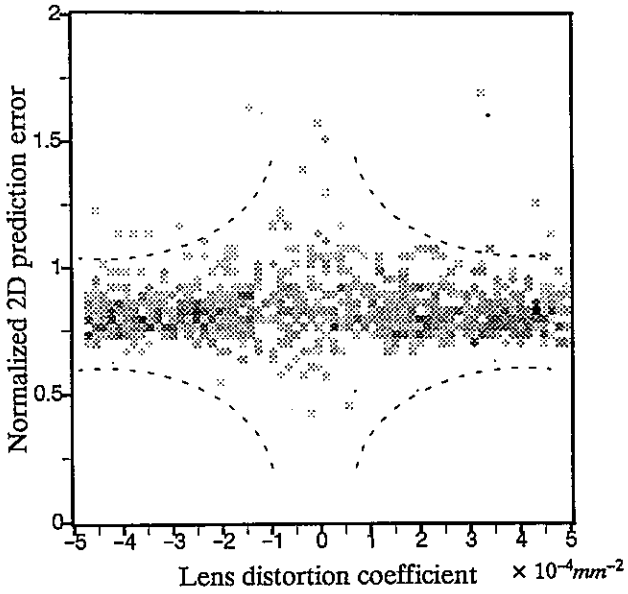


Fig. 14.

the theoretic bound, i.e., the normalized error $\lesssim 1$ with high probability. Still, there are some points which exceed the theoretic bound. This is partially because of the violation of the assumption (a.3) in section IV. Another reason is that $\epsilon_n^2$ is an expectation value, not an upper bound. Figs. 8 – 14 show the distribution of the random trials with the normalized 2D prediction error as vertical axis and the parameter we are interested in as horizontal axis, where

darkness represents the occurrence frequency of the random trials. Some parameters do not show strong relation to the 2D prediction error, these are the effective focal length, $f$, the principle point, $(u_0, v_0)$, and depth of the working volume, see Fig. 8 – 11. In Fig. 14, we can see that when the lens distortion is very small, i.e. $|\kappa| \approx 0$, the effects of the 2D observation noise dominates, and the $\epsilon_{Bound}$ is more of an error expectation than of an upper bound. Therefore, the normalized error varies in a larger extent. Figs. 12 and 13 show that when the number of calibration points is small or the 2D observation noise is large, the approximate bound tends to be violated. At the beginning we expect that the smaller the depth of the working volume is (to let the assumption (a.3) be true), the more correct the bound is. But due to the effects of the random 2D observation noise (recall that we need noncoplanar points for calibration, and the smaller the depth of the working volume the more singularity the calibration problem tends to, since the calibration points tens to be on the same plane), we do not see this phenomenon in Fig. 11.

The third experiment test the bound by a real experiment. With a PULNiX TM–745E camera, and an ITI Series 151 frame grabber, we took 21 images of a moving calibration plate having 25 calibration points on it, which is mounted on a translation stage. One image was taken each time the translation stage was moved toward the camera by 25 minimeters. A typical image is shown in Fig. 16. Thus we have $21 \times 25 = 525$ pairs of 2D-3D coordinates of points. The image coordinates of the center for each circle is estimated, with an error of about 0.1 pixel. For $N_{calib} = 10, 20, 30, \ldots$ and 200, we randomly choose $N_{calib}$ points from the 525 2D-3D pairs to calibrate the camera and use all remaining points to test the calibrated parameters. The above random trials are repeated ten times to obtain ten sets of the 2D prediction error. Fig. 15 shows the ten sets of data and two predicted bounds based on two different effective image sizes (here $\kappa = 0.00035$ minimeter$^{-2}$). Since all the calibration and test points are distributed in the central part of the image, whose size is roughly of 355 by 300 pixels (see Fig. 16), the bound calculated with this image size is much closer to the experimental

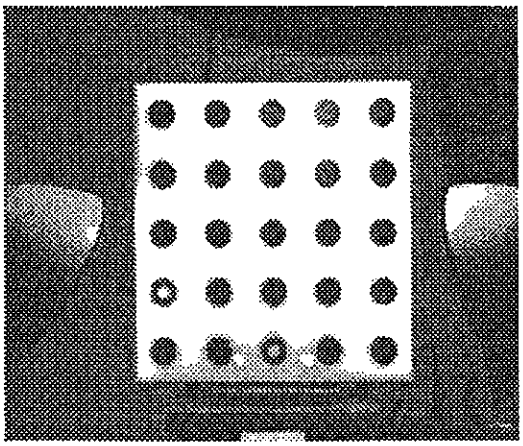

Fig. 16. A typical image of the calibration plate containing 25 calibration points used in the real experiments.

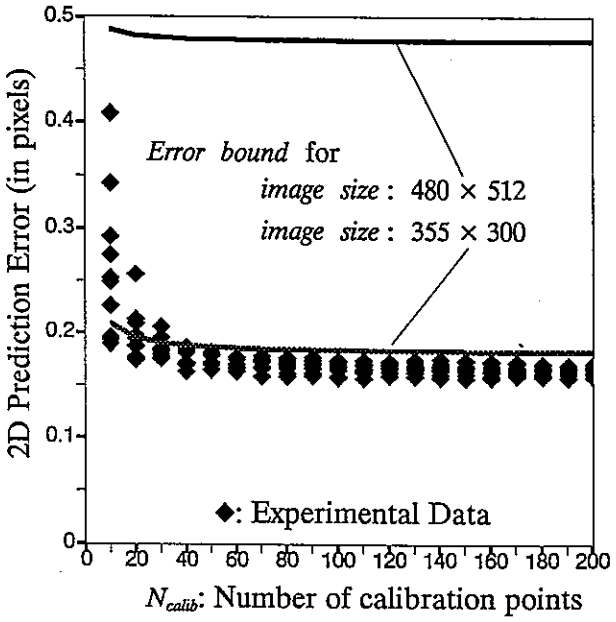


Fig. 15. Real experiments

results. To use every pixels in the $480 \times 512$ image, the error bound will be approximately three times of the experimental results.

## VI. CONCLUSIONS

In this paper, we have derived an approximate upper bound for the 2D prediction error. The effects of both the radial lens distortion and the 2D observation noise are considered. This bound was tested by computer simulations and real experiments which show that the upper bound is quite tight, i.e., it is close to the experimental results and still bounds almost all of them from above. For 3D applications, e.g., stereo vision, it is of great importance to determine the accuracy of 3D position estimation. Knowing the 2D prediction error, the 3D position error can be derived as in [1]. Thus, the error bound can be used as a criterion to decide whether the linear camera model is sufficient or not, for a specific application. In the following, a general guide line is provided for using this error bound:

1). Determine the acceptable 2D prediction error and 3D angular error. If the specified error bound is given as the 3D angular error, then equation (11) is used to translate it to the 2D prediction error. For convenience, let us denote this specified 2D error bound as $\epsilon_{spec}$.

2). Calculate the approximate error bound, $\epsilon_{Bound}$, by equation (43) according to the parameters of the equipments to be used.

3). If $\epsilon_{spec} > \epsilon_{Bound}$ then it is good enough to use the linear camera model.

4). If $\epsilon_{spec} < \epsilon_{Bound}$ then try to reduce $\epsilon_n$ in equation (43) as much as possible, by making the feature extraction more accurate (reduce $\sigma$) and increase the number of calibration points. Check if this process brings the theoretic bound, $\epsilon_{Bound}$, to the value smaller than the specified one, $\epsilon_{spec}$.

5). If $\epsilon_{Bound}$ still can not meets the requirement after the reduction of $\epsilon_n$ in step 4), then try to reduce the effective size of the image to an acceptable level (see equation (42)).

6). If the efforts in step 4) and 5) fail to reduce $\epsilon_{Bound}$ such that $\epsilon_{spec} > \epsilon_{Bound}$, then a nonlinear camera model should be considered in the camera calibration procedure as in [3] [8] [12] [13].

A linear camera model is always the first consideration of engineers. Not only will it simplify the camera calibration procedure, but will it make the subsequent processing easier (e.g. eliminating the need of geometric correction). This paper provides a tool for making decisions based on the trade–off between accuracy and efficiency.

## APPENDIX I.

As shown in Fig. 17(a) and (b), the average distance from the image point to the estimated lens center, $d_a$, and the average pixel spacing, $\delta_a$, can be computed through the following equations:

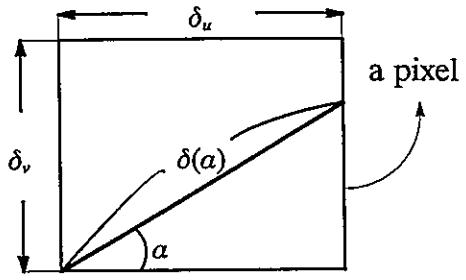


Fig. 17(a). The average pixel spacing.

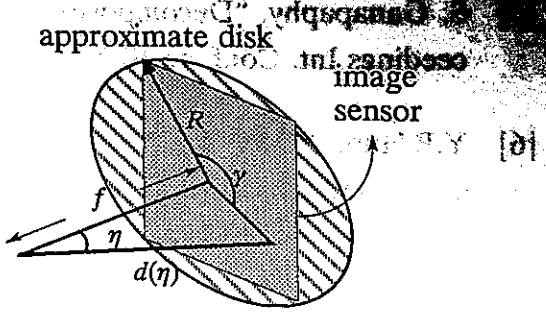


Fig. 17(b). The average distance to the estimated lens center.

$$d_a \equiv \frac{1}{2\pi\eta_r}\int_0^{\eta_r}\int_0^{2\pi} r(\eta)d\gamma d\eta = \frac{f}{\eta_r}\ln|sec(\eta_r)+\tan(\eta_r)|\,, \tag{44}$$

$$\delta_a \equiv \frac{2}{\pi}\int_0^{\frac{\pi}{2}}\delta(a)da = \frac{2}{\pi}[\delta_u \ln|\sec(a_u)+\tan(a_u)| + \delta_v \ln|\sec(a_v)+\tan(a_v)|]\,, \tag{45}$$

where $\eta_r \equiv \tan^{-1}(\frac{R}{f})$,

$R = \sqrt{(\delta_u u_{\max})^2 + (\delta_v v_{\max})^2}$,

$a_u \equiv \tan^{-1}(\frac{\delta_v}{\delta_u})$,

and $a_v \equiv \tan^{-1}(\frac{\delta_u}{\delta_v})$.

## REFERENCES


[1] S.D. Blostein, T.S. Hung, "Error Analysis in Stereo Determination of 3–D Point Positions," IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI–9, NO. 6, Nov., 1987, pp. 752–765.

[2] R.O. Duda, P.E. Hart, Pattern Recognition and Scene Analysis, Wiley, New York, 1973.

[3] W. Faig, "Calibration of Close–Range Photogrammetry Systems: Mathematical Formulation," Photogrammetric Engineering and Remote Sensing, Vol. 41, No. 12, 1975, pp. 1479–1486.

[4] O.D. Faugeras, G. Toscani, "The Calibration Problem for Stereo," Proceedings Conf. on Computer Vision and Pattern Recognition, 1986, pp. 15–20.

[5] S. Ganapaphy, "Decomposition of Transformation Matrices for Robot Vision," Proceedings Int. Conf. on Robotics and Automation, 1984, pp. 130–139.

[6] Y.P. Hung, "Three Dimensional Surface Reconstruction Using a Moving Camera A Model–Based Probabilistic Approach," Ph.D dissertation, Division of Engineering, Brown University, Providence, R.I.; also, Technical Report LEMS–63, 1989.

[7] Y.P. Hung, S.W. Shih, "When Should We Consider Lens Distortion in Camera Calibration," IAPR Workshop on Machine Vision Applications, Tokyo, Nov. 1990, pp. 367–370.

[8] S.W. Shih, Y.P. Hung, W.S. Lin, "An Efficient and Accurate Camera Calibration Technique for 3D computer Vision," to appear in Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision VI, November 1991; also Technical Report TR–91–018, Institute of Information Science, Academia Sinica, Taipei.

[9] S.W. Shih, Y.P. Hung, W.S. Lin, "Accuracy Assessment on Camera Calibration Method not Considering Lens Distortion," Technical Report TR–92–001, Institute of Information Science, Academia Sinica, Taipei.

[10] T.M. Strat, "Recovering the Camera Parameters from a Transformation Matrix," DARPA Image Understanding Workshop, 1984.

[11] I. Sutherland, "Three–Dimensional Data Input by Tablet," Proceedings of the IEEE, Vol. 62, No. 4, 1974, pp. 453–461.

[12] R.Y. Tsai, "A Versatile Camera Calibration Technique for High–Accuracy 3D Machine Vision Metrology Using Off–the–Shelf TV Cameras and Lenses," IEEE Journal of Robotics and Automation, Vol. RA–3 No. 4, 1987, pp. 323–344.

[13] J. Weng, P. Cohen, M. Herniou, "Calibration of Stereo Cameras Using a Non–linear Distortion Model," Proc. 10th Inter. Conf. on Pattern Recognition, 1990, pp. 246–253.