Pergamon

Pattern Recognition, Vol. 28, No. 3, pp. 447–461, 1995
Elsevier Science Ltd

0031-3203/95 $9.50 + .00

0031-3203(94)00107-3

WHEN SHOULD WE CONSIDER LENS DISTORTION IN CAMERA CALIBRATION

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.

(*Received* 23 *November* 1993; *in revised form* 3 *August* 1994; *received for publication* 22 *August* 1994)

Abstract—This work investigates the effect of neglecting lens distortion in camera calibration, and presents a theoretical analysis of the calibration accuracy. We derived an approximate upper envelope for the 2D prediction error, which 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 envelope 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.

Camera calibration Lens distortion Accuracy assessment Error bound
Camera model

1. 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 β, 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 β, is called (geometric) camera calibration.

The techniques for camera calibration can be classified into two categories: one that considers lens distortion,[1–4] and one that neglects lens distortion.[5–8] A typical linear technique that does not consider lens distortion is the one estimating the perspective transformation matrix H.[6,8] The estimated H can be used *directly* for forward and backward 3D–2D projection. If necessary, given the estimated H, the geometric camera parameters β can be easily determined.[6,7,9]

The Faig method[1] 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'.[3] His method involves a direct solution for most of the calibration parameters and some iterative solution for the remaining parameters. Some drawbacks of the Tsai method are pointed out in reference (4). Our experiences[10] also showed that the Tsai method can be worse than the simple linear method of[8] if lens distortion is relatively small.

Recently, Weng showed some experimental results using a two-step method.[4] 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 the Tsai method according to our experiments.

We have also developed a fast and accurate technique for calibrating a camera, with lens distortion, by

* Author to whom correspondence should be addressed.

solving linear equations.[2] Instead of using nonlinear optimization techniques, the estimation of radial lens distortion coefficient is transformed into an eigenvalue problem of an 8 × 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 the Tsai method.

However, considering lens distortion will not only complicate the camera calibration procedure, but also complicate the subsequent on-line processing (though not formidable) 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 is it worth all the trouble to consider lens distortion?' This work represents an effort towards the answer of this question.

2. 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 $\boldsymbol{r}_O = (x_O, y_O, z_O)^t$ be its coordinates, in millimeters, with respect to a fixed object coordinate system (OCS). Let the camera coordinate system (CCS), also in millimeters, have its x–y plane parallel to 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), with its origin located at the lens center and its z axis aligned with the optical axis of the lens (see Fig. 1). Let $\boldsymbol{r}_C = (x_C, y_C, z_C)^t$ be 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 $s_I = (u_I, v_I)^t$ 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 set at *the center of the frame memory coordinate* [e.g. the origin of the ICS is set at (256, 256) for a 512 by 512 image].

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

2.1. *Translation and rotation from the OCS to the CCS*

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

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

i.e.

$$\begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_O \\ y_O \\ z_O \\ 1 \end{bmatrix}, \tag{2}$$

where tilde ($\sim$) denotes homogeneous coordinates,[11] $\boldsymbol{t}_C^O = (t_1\, t_2\, t_3)^t$ is a translation vector, and $\boldsymbol{R}_C^O$ is a 3 × 3 rotation matrix determined by the three Euler angles, ϕ, θ, ψ, rotating about the z, y, z axes sequentially.

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

Let f be the distance between CCS and ICS plane as shown in Fig. 1, and referred to as the 'effective focal length'. Let $s_F = (u_F, v_F)^t$ be the 2D coordinates (in millimeters) of the undistorted image point Q lying on

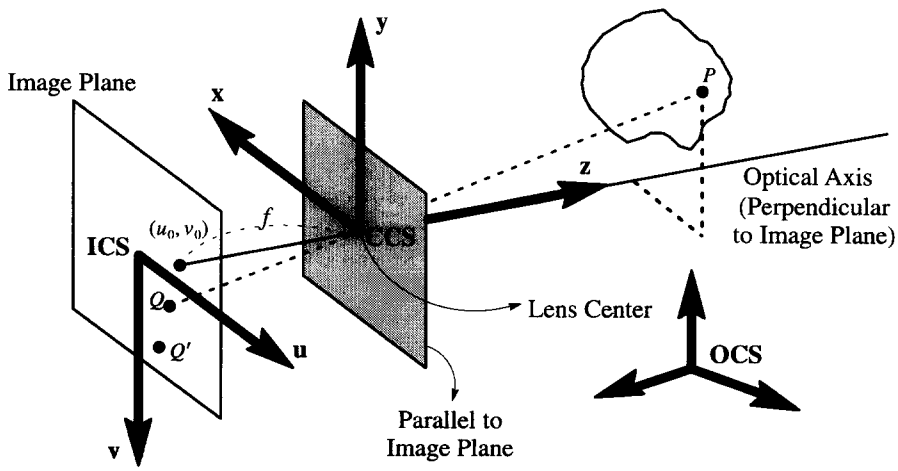

Fig 1. Pinhole camera model with lens distortion, where P is a 3D object point, Q and Q' are its undistorted and distorted image points, respectively. OCS—Object Coordinate System (3D). CCS—Camera Coordinate System (3D). ICS—Computer Image Coordinate System (2D).

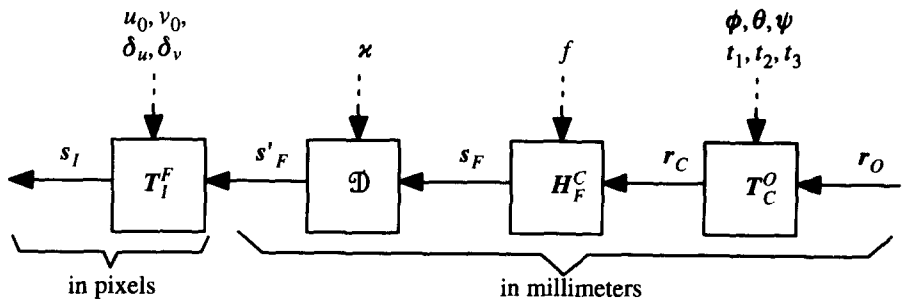

Fig. 2. Relation between different transformation matrices.

the image plane. Then, we have

$$u_F = f\frac{x_C}{z_C}, \quad v_F = f\frac{y_C}{z_C}. \tag{3a}$$

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

$$\tilde{s}_F = \boldsymbol{H}_F^C \tilde{\boldsymbol{r}}_C \quad \text{with} \quad \boldsymbol{H}_F^C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix}. \tag{3b}$$

2.3. *Lens distortion from* Q *to* Q′

For practical reasons as mentioned in Tsai,[3] we consider only the first term of the radial lens distortion, i.e.

$$s_F = (1 - \kappa \|s'_F\|^2) s'_F, \tag{4}$$

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

2.4. *Scaling and translation of 2D image coordinates*

The transformation from s'_F (in millimeters) to s_I (in pixels) involves: (i) scaling from millimeter to pixels, and (ii) translation due to misalignment of the sensor array with the optical axis of the lens. Hence,

$$\tilde{s}_I = \boldsymbol{T}_I^F \tilde{s}'_F \quad \text{with} \quad \boldsymbol{T}_I^F = \begin{bmatrix} 1/\delta_u & 0 & u_0 \\ 0 & 1/\delta_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

where δ_u and δ_v are the horizontal and vertical pixel spacing (millimeter/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 δ_v is not included here because it is a known parameter when we use a solid state camera—otherwise, only the ratios f/δ_u and f/δ_v can be determined. Combining (1), (2), (3) and (4), we have

$$(1 - \kappa\varrho^2)(u_I - u_0)\delta_u = f\frac{x_O r_1 + y_O r_2 + z_O r_3 + t_1}{x_O r_7 + y_O r_8 + z_O r_9 + t_3} \tag{6a}$$

$$(1 - \kappa\varrho^2)(v_I - v_0)\delta_v = f\frac{x_O r_4 + y_O r_5 + z_O r_6 + t_2}{x_O r_7 + y_O r_8 + z_O r_9 + t_3} \tag{6b}$$

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 $\mathcal{D}$ is an identity operator, see Fig. 2), the relationship between $\boldsymbol{r}_O$ and $\boldsymbol{s}_I$ can be expressed as a linear transformation by combining (1), (2) and (4):

$$\tilde{s}_I = \boldsymbol{H}\tilde{\boldsymbol{r}}_O \quad \text{i.e.} \quad \begin{bmatrix} u_I \cdot w \\ v_I \cdot w \\ w \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ h_5 & h_6 & h_7 & h_8 \\ h_9 & h_{10} & h_{11} & 1 \end{bmatrix} \begin{bmatrix} x_O \\ y_O \\ z_O \\ 1 \end{bmatrix}, \tag{7}$$

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

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

3. A LINEAR METHOD FOR CAMERA CALIBRATION

Given a set of 3D calibration points and their corresponding 2D image coordinates, the problem is to estimate β, the parameters of our camera model. Instead of estimating β directly, we first estimate the composite parameters $\mathbf{h}$ [as described following equation (10)], then the composite parameters can be decomposed into β by methods described in Ganapaphy,[9] Hung[6] and Strat.[7]

From equation (7), the derivation of a linear method for camera calibration is quite simple [refer to references (6), (7) and (9)]. However, to observe the effects of lens distortion on camera calibration, it is necessary to derive the linear method in a different way.

Assuming that we have N_{calib} pairs of 2D–3D calibration points, from (6a) and (6b) it can be shown that (see Appendix A)

$$\boldsymbol{Ap} + \boldsymbol{Bq} + \kappa \boldsymbol{Cq} = 0. \tag{8}$$

where

$$\boldsymbol{A} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{2N_{\text{calib}} \times 8},$$

$$\boldsymbol{B} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_j x_j & -u_j y_j & -u_j z_j & -u_j \\ -v_j x_j & -v_j y_j & -v_j z_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{2N_{\text{calib}} \times 4},$$

$$\boldsymbol{C} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ (u_j - u_0)\varrho_j^2 x_j & (u_j - u_0)\varrho_j^2 y_j & (u_j - u_0)\varrho_j^2 z_j & (u_j - u_0)\varrho_j^2 \\ (v_j - v_0)\varrho_j^2 x_j & (v_j - v_0)\varrho_j^2 y_j & (v_j - v_0)\varrho_j^2 z_j & (v_j - v_0)\varrho_j^2 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{2N_{\text{calib}} \times 4},$$

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

and

$$\boldsymbol{p} \equiv \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \end{bmatrix} \Big/ t_3 = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_8 \end{bmatrix}_{8 \times 1}, \quad \boldsymbol{q} \equiv \boldsymbol{P}_3/t_3 \equiv \begin{bmatrix} r_7/t_3 \\ r_8/t_3 \\ r_9/t_3 \\ 1 \end{bmatrix} = \begin{bmatrix} h_9 \\ h_{10} \\ h_{11} \\ 1 \end{bmatrix}.$$

From the above definition of $\boldsymbol{P}_3$, we have, using equation (2),

$$[x_j \; y_j \; z_j \; 1]\boldsymbol{P}_3 = z_{Cj}, \tag{9}$$

where z_{Cj} is the z-component of the coordinates of the jth calibration point in the camera coordinate system.

Suppose we have a distortionless lens, i.e. $\kappa = 0$, then equation (8) can be simplified to

$$\boldsymbol{A}\boldsymbol{p} + \boldsymbol{B}\boldsymbol{q} = [\boldsymbol{A} \;\; \boldsymbol{B}]\begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix}$$

$$= [\boldsymbol{A} \;\; \boldsymbol{B}' \;\vdots\; -\boldsymbol{b}]\begin{bmatrix} \boldsymbol{h} \\ 1 \end{bmatrix} = \boldsymbol{e} \approx 0, \tag{10}$$

where $\boldsymbol{B}'$ is the matrix space obtained by removing the last column of $\boldsymbol{B}$, $\boldsymbol{b} \equiv [\cdots u_j v_j \cdots]^t$ (i.e. $[\boldsymbol{B}' \;\vdots\; -\boldsymbol{b}] = \boldsymbol{B}$) and $\boldsymbol{h} = [h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8 h_9 h_{10} h_{11}]^t$. Notice that the small error vector, $\boldsymbol{e}$, in equation (10) 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, $\|\boldsymbol{e}\|^2$, with respect to $\mathbf{h}$:

$$\|\boldsymbol{e}\|^2 = \|\Lambda\boldsymbol{h} - \boldsymbol{b}\|^2, \tag{11}$$

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

The optimal solution of (11) is well known to be

$$\hat{\boldsymbol{h}} = (\Lambda^t\Lambda)^{-1}\Lambda^t\boldsymbol{b}, \tag{12}$$

if there are more than *six noncoplanar calibration points*. The estimated composite parameters can be further decomposed into β, when necessary.

4. ACCURACY ASSESSMENT

In this section we will derive an approximate error envelope for the linear calibration method shown in Section 3, 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 envelope is based on the following assumptions:

Assumption 1. The 3D positions of the calibration points are known exactly. In practice, the positions of control points can be determined with a high accuracy, much higher than that represented by the size of the back-projected pixel at the corresponding depth. Furthermore, the 3D position error can always be transformed to an equivalent 2D measuring error.

Assumption 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 are independent and identically distributed Gaussian with zero mean and the variance, σ^2.

Assumption 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 many computer vision applications, since the depth of field for a practical camera is usually limited to a small range comparing to the object distance.

4.1. *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. Consider Fig. 3, $\hat{O}$ is the estimated optical center. Q is the observed 2D image point corresponding to the 3D test point P. $\hat{Q}$ is obtained by projecting the 3D test point P onto the image plane using the estimated camera parameters $\hat{\beta}$. Because of the estimation error in $\hat{\beta}$, the predicted 2D image point $\hat{Q}$ will not be at the same location as the observed image point Q. By using the estimated camera parameter $\hat{\beta}$, we can also compute the 3D ray, $\hat{O}Q$, by back-projecting a 3D line from $\hat{O}$ through Q to 3D space. In general, $\hat{O}Q$ does not pass through the test point P. Based on the above notations, we can define two error measures as follows.

(i) the *3D angular error*, i.e. the angle (in degree) $\measuredangle P\hat{O}Q$.

(ii) the 2D *prediction error*, i.e. the image distance (in pixels) between the Q and $\hat{Q}$.

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

$$\text{3D angular error} \approx \text{2D prediction error} \times \frac{\delta_a}{d_a}, \tag{13}$$

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

In the following, we are going to give the intuition of the error function (11). 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.

$$\hat{u} \equiv \frac{\hat{h}_1 x + \hat{h}_2 y + \hat{h}_3 z + \hat{h}_4}{\hat{h}_9 x + \hat{h}_{10} y + \hat{h}_{11} z + 1}, \tag{14a}$$

and

$$\hat{v} \equiv \frac{\hat{h}_5 x + \hat{h}_6 y + \hat{h}_7 z + \hat{h}_8}{\hat{h}_9 x + \hat{h}_{10} y + \hat{h}_{11} z + 1}, \tag{14b}$$

Notice that the error function (11) is equivalent to the following equation (see Appendix C)

$$\|\boldsymbol{e}\|^2 = \|\Lambda \boldsymbol{h} - \boldsymbol{b}\|^2 = \sum_{j=1}^{N_{\text{calib}}} \left[(u_j - \hat{u}_j)^2 \left[\frac{\hat{z}_{Cj}}{\hat{t}_3} \right]^2 + (v_j - \hat{v}_j)^2 \left[\frac{\hat{z}_{Cj}}{\hat{t}_3} \right]^2 \right], \tag{15}$$

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 (15), 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 3 and equation (15), we have

$$\|\boldsymbol{e}\|^2 \approx \left\{ \sum_{j=1}^{N_{\text{calib}}} [(u_j - \hat{u}_j)^2 + (v_j - \hat{v}_j)^2] \right\} \times \left(\frac{z_C}{t_3} \right)^2, \tag{16}$$

which amounts to say that equation (12) is the *optimal solution* that minimizes the 2D prediction error.

4.2. *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 (10) can be written as

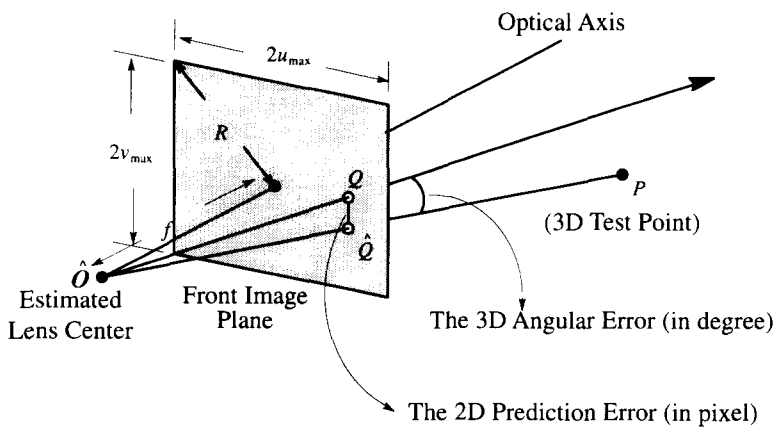

Fig. 3. The error measures used in this paper.

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \end{bmatrix} + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_{\text{true}j}x_j & -u_{\text{true}j}y_j & -u_{\text{true}j}z_j & -u_{\text{true}j} \\ -v_{\text{true}j}x_j & -v_{\text{true}j}y_j & -v_{\text{true}j}z_j & -v_{\text{true}j} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \boldsymbol{P}_3 = 0. \tag{17}$$

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

$$u = u_{\text{true}} + n_u \quad \text{and} \quad v = v_{\text{true}} + n_v, \tag{18}$$

where $(u_{\text{true}}, v_{\text{true}})$ is the true image coordinate (noise free) and (u, v) is the measured one (noisy). Substituting u_{true} and v_{true} in (18) into (17), we have

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \end{bmatrix} + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_jx_j & -u_jy_j & -u_jz_j & -u_j \\ -v_jx_j & -v_jy_j & -v_jz_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \boldsymbol{P}_3 + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ n_{uj}x_j & n_{uj}y_j & n_{uj}z_j & n_{uj} \\ n_{vj}x_j & n_{vj}y_j & n_{vj}z_j & n_{vj} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \boldsymbol{P}_3 = 0. \tag{19}$$

Substituting (9) into (19) and dividing (19) by t_3, it follows that

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \boldsymbol{p} + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_jx_j & -u_jy_j & -u_jz_j & -u_j \\ -v_jx_j & -v_jy_j & -v_jz_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \boldsymbol{q} + \begin{bmatrix} \vdots \\ n_{uj}z_{Cj}/t_3 \\ n_{vj}z_{Cj}/t_3 \\ \vdots \end{bmatrix} = 0. \tag{20}$$

Comparing equations (10) and (20), we have

$$\boldsymbol{e} = -\begin{bmatrix} \vdots \\ n_{uj}z_{Cj}/t_3 \\ n_{vj}z_{Cj}/t_3 \\ \vdots \end{bmatrix} \approx -\begin{bmatrix} \vdots \\ n_{uj} \\ n_{vj} \\ \vdots \end{bmatrix} z_C/t_3. \tag{21}$$

Using equations (10) and (11), we have

$$\Lambda \boldsymbol{h} = \boldsymbol{b} + \boldsymbol{e} \equiv \boldsymbol{b}_T, \tag{22}$$

where $\boldsymbol{b}_T$ is true value of the noise corrupted vector $\boldsymbol{b}$.

Ideally, the test points are noise free, and 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 tested by using the noise free calibration points. Let us denote the expectation of the root mean square 2D prediction error as ε_n, then we have

$$\varepsilon_n^2 \equiv \frac{1}{N_{\text{calib}}} E\left\{ \sum_{j=0}^{N_{\text{calib}}} [(u_{\text{true}j} - \hat{u}_j)^2 + (v_{\text{true}j} - \hat{v}_j)^2] \right\}. \tag{23}$$

By Assumption 3, we have

$$N_{\text{calib}} \times \left(\frac{z_C}{t_3}\right)^2 \times \varepsilon_n^2 \approx E\left\{ \sum_{j=0}^{N_{\text{calib}}} \left[(u_{\text{true}j} - \hat{u}_j)^2 \left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right)^2 + (v_{\text{true}j} - \hat{v}_j)^2 \left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right)^2 \right] \right\}. \tag{24}$$

From equations (22) and (15), equation (24) can be further simplified as

$$N_{\text{calib}} \times \left(\frac{z_C}{t_3}\right)^2 \times \varepsilon_n^2 = E[\|\boldsymbol{b}_T - \Lambda \hat{\boldsymbol{h}}\|^2]. \tag{25}$$

Since all the measurements in equation (22) are noise-free, it is obvious that the solution of (22) obtained by pseudo inverse is equal to the true value of $\boldsymbol{h}$, and thus it has zero residual error, i.e.

$$\begin{aligned} \boldsymbol{b}_T - \Lambda \boldsymbol{h} &= \boldsymbol{b}_T - \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\boldsymbol{b}_T \\ &= (\boldsymbol{b} + \boldsymbol{e}) - \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t(\boldsymbol{b} + \boldsymbol{e}) = \boldsymbol{0}. \end{aligned} \tag{26}$$

From equation (12) and (26), we have

$$\begin{aligned} \|\boldsymbol{b}_T - \Lambda \hat{\boldsymbol{h}}\|^2 &= \boldsymbol{b} + \boldsymbol{e} - \Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\boldsymbol{b}\|^2 \\ &= \|\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\boldsymbol{e}\|^2. \end{aligned} \tag{27}$$

The expectation of equation (27) is (see Appendix D)

$$E\|\boldsymbol{b}_T - \Lambda \hat{\boldsymbol{h}}\|^2 = \frac{11\sigma^2 z_C^2}{t_3^2}. \tag{28}$$

Taking average of the above equation over the N_{calib} points and dividing it with the constant z_C^2/t_3^2 [see equation (25)], we have the *expectation* of the average square 2D prediction error

$$\varepsilon_n^2 \equiv \frac{11\sigma^2}{N_{\text{calib}}} \quad \text{(in pixels)}. \tag{29}$$

4.3. *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 subsection. Another is due to the improper modeling of the camera, which will be dealt with in this subsection.

Notice that the linear calibration method will minimize the 2D prediction error, [see equation (16)], subject to the distortion free camera model and providing that the assumption 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 envelope, since the optimal solution will always have a smaller error. *When more approximations are used in the derivation of the approximate solution, the envelope will be more conservative.* Because of that the approximate solution deviates more from the optimal solution. The derivation of the upper envelope of the modeling error is described below.

4.3.1. *Find an approximate relation between the estimated composite parameters and the true one.* Rewrite (8) as follows:

$$\Lambda \boldsymbol{h} = \boldsymbol{b} - \kappa \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \\ (u_j - u_0)\varrho_j^2 x_j & (u_j - u_0)\varrho_j^2 y_j & (u_j - u_0)\varrho_j^2 z_j & (u_j - u_0)\varrho_j^2 & \dfrac{\boldsymbol{P}_3}{t_3} \\ (v_j - v_0)\varrho_j^2 x_j & (v_j - v_0)\varrho_j^2 y_j & (v_j - v_0)\varrho_j^2 z_j & (v_j - v_0)\varrho_j^2 & \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix}. \tag{30}$$

Since $z_{Cj} = [x_j \; y_j \; z_j \; 1]\boldsymbol{P}_3$ and $z_{C1} \approx z_{C2} \approx \cdots \approx z_{CN} \approx z_C$ (see Assumption 3), we have

$$\Lambda \boldsymbol{h} \approx \boldsymbol{b} - \kappa \begin{bmatrix} \vdots \\ (u_j - u_0)\varrho_j^2 \dfrac{z_{Cj}}{t_3} \\ (v_j - v_0)\varrho_j^2 \dfrac{z_{Cj}}{t_3} \\ \vdots \end{bmatrix} \approx \boldsymbol{b} - \kappa \frac{z_C}{t_3} \begin{bmatrix} \vdots \\ (u_j - u_0)\varrho_j^2 \\ (v_j - v_0)\varrho_j^2 \\ \vdots \end{bmatrix}.$$

In general, the principle point (u_0, v_0), is negligible comparing to the image points, (u_j, v_j), we have (see the definition of ICS)

$$\Lambda \boldsymbol{h} \approx \boldsymbol{b} - \kappa \frac{z_C}{t_3} \begin{bmatrix} \vdots \\ (u_j - u_0)\varrho_j^2 \\ (v_j - v_0)\varrho_j^2 \\ \vdots \end{bmatrix} \approx \boldsymbol{b} - \kappa \frac{z_C}{t_3} \begin{bmatrix} \vdots \\ (u_j)\varrho_j^2 \\ (v_j)\varrho_j^2 \\ \vdots \end{bmatrix}. \tag{31}$$

Now suppose that ϱ_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 (31), yields

$$\Lambda \boldsymbol{h} \approx (1 - \kappa M z_C / t_3)\boldsymbol{b}. \tag{32}$$

From (32), we have

$$\boldsymbol{h}_{\text{true}} \approx (1 - \kappa M z_C/t_3)(\Lambda^t \Lambda)^{-1} \Lambda^t \boldsymbol{b} = (1 - \kappa M z_C/t_3)\hat{\boldsymbol{h}}, \tag{33}$$

where $\boldsymbol{h}_{\text{true}}$ is an exact solution of equation (30)

4.3.2. *Find the relation of the true image point* $(u_{\text{true}}, v_{\text{true}})$ *and its undistorted image point* (u_{ud}, v_{ud}). [Refer to Fig. 4(a).] Using the estimated parameters $\hat{\boldsymbol{h}}$ the predicted image point $(\hat{u}, \hat{v})$ (in pixels) is defined in equation (14). Also, if the true parameters $\boldsymbol{h}_{\text{true}}$ were used, then the undistorted image point (u_{ud}, v_{ud}) (in pixels) will be

$$u_{ud} = \frac{h_1 x + h_2 y + h_3 z + h_4}{h_9 x + h_{10} y + h_{11} z + 1}, \tag{34a}$$

$$v_{ud} = \frac{h_5 x + h_6 y + h_7 z + h_8}{h_9 x + h_{10} y + h_{11} z + 1}, \tag{34b}$$

where the subscript '*ud*', 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 (5), we have

$$\boldsymbol{s}_F' = \begin{bmatrix} (u - u_0)\delta_u \\ (v - v_0)\delta_v \end{bmatrix} \quad \text{and} \quad \boldsymbol{s}_F = \begin{bmatrix} (u_{ud} - u_0)\delta_u \\ (v_{ud} - v_0)\delta_v \end{bmatrix}, \tag{35}$$

In practice, $\kappa\varrho^2 \ll 1$ (e.g. $\kappa = 0.00035\,\text{mm}^{-2}$, $\varrho_{\max}^2 = 25.80\,\text{mm}^2$, and $\kappa\varrho_{\max}^2 = 0.01$). Therefore from equation (4), it follows that

$$\begin{bmatrix} (u_{\text{true}} - u_0) \\ (v_{\text{true}} - v_0) \end{bmatrix} = 1/(1 - \kappa\varrho^2) \begin{bmatrix} (u_{ud} - u_0) \\ (v_{ud} - v_0) \end{bmatrix} \approx (1 + \kappa\varrho^2) \begin{bmatrix} (u_{ud} - u_0) \\ (v_{ud} - v_0) \end{bmatrix}. \tag{36}$$

4.3.3. *Find the relation of the predicted 2D image point* $(\hat{u}, \hat{v})$ *and its true undistorted image point* (u_{ud}, v_{ud}). [Refer to Fig. 4(a) and (b).] By equations (14), (33) and (34), it can be shown that (see Appendix E).

$$\begin{bmatrix} \hat{u} - u_0 \\ \hat{v} - v_0 \end{bmatrix} \approx (1 + \kappa M) \begin{bmatrix} u_{ud} - u_0 \\ v_{ud} - v_0 \end{bmatrix}. \tag{37}$$

4.3.4. *Find the error of the predicted 2D image point* $(\hat{u}, \hat{v})$ *with respect to its true image point* $(u_{\text{true}}, v_{\text{true}})$. [Refer to Fig. 4(a) and (b).] By (36) and (37), the error of the predicted 2D image point, $(\hat{u}, \hat{v})$, with respect to

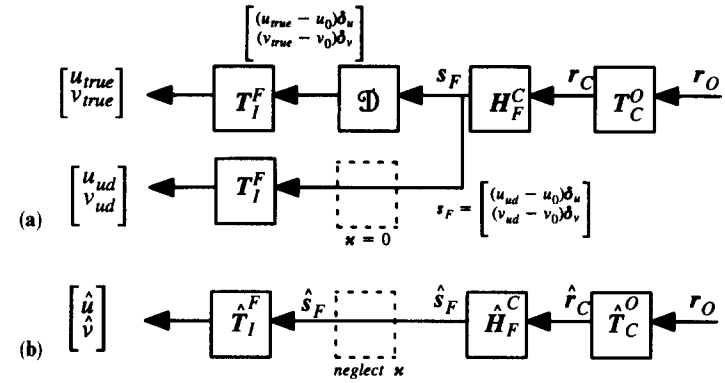

Fig. 4. (a) This figure shows the relation between the true image point $[u_{true}, v_{true}]^t$ and its undistorted image point, $[u_{true}, v_{true}]^t$, when the true parameters are known. (b) The 2D predicted image $[\hat{u}, \hat{v}]$ is computed using the camera parameters estimated using linear calibration method without considering lens distortions.

the corresponding true image point, (u_{true}, v_{true}), is:

$$\begin{bmatrix}(u_{true}-\hat{u})\\(v_{true}-\hat{v})\end{bmatrix}$$

$$=\begin{bmatrix}(u_{true}-u_0)-(\hat{u}-u_0)\\(v_{true}-v_0-(\hat{v}-v_0)\end{bmatrix}$$

$$\approx\begin{bmatrix}(u_{ud}-u_0)(1+\kappa\varrho^2)-(u_{ud}-u_0)(1+\kappa M)\\(v_{ud}-v_0)(1+\kappa\varrho^2)-(v_{ud}-v_0)(1+\kappa M)\end{bmatrix}$$

$$=\begin{bmatrix}\kappa(u_{ud}-u_0)(\varrho^2-M)\\\kappa(v_{ud}-v_0)(\varrho^2-M)\end{bmatrix}. \tag{38}$$

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, δ_u, δ_v, respectively. Let U and V denote $(u_{ud}-u_0)$ and $(v_{ud}-v_0)$, respectively. As explained before, $\kappa\varrho^2 \ll 1$. Hence, from (38), the 2D square error becomes

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

$$=\kappa^2\varrho_{ud}^2(\varrho^2-M)^2, \tag{39}$$

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

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

$$\varepsilon_M^2=\frac{1}{u_{max}v_{max}}\int_0^{u_{max}}\int_0^{v_{max}}\kappa^2(\varrho^2-M)^2((U\delta_u)^2+(V\delta_v)^2)\,dU\,dV,$$

$$\approx\frac{1}{u_{max}v_{max}}\int_0^{u_{max}}\int_0^{v_{max}}\kappa^2(\varrho_{ud}^2-M)^2\varrho_{ud}^2\,dU\,dV, \tag{40}$$

where ε_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

$$\varepsilon_M^2 \approx \frac{1}{\pi R^2}\int_0^R\int_0^{2\pi}\kappa^2(\hat{\varrho}^2-M)^2\hat{\varrho}^2\,d\theta\,d\varrho$$

$$=\kappa^2\left[\frac{R^6}{9}-\frac{2R^4M}{6}+\frac{R^2M^2}{4}\right] \quad \text{(in millimeters)} \tag{41}$$

Minimize (41) subject to M, we have

$$M=2R^2/3, \quad \text{and} \quad \varepsilon_M^2=\kappa^2R^6/36. \tag{42}$$

Recall that we claim that ϱ_j^2 can be approximated to a constant, which yields equation (33), but in practice, this is usually not a good approximation. Besides, since $\hat{\boldsymbol{h}}$ is the optimal solution that minimize ε_M^2, and $(1+\kappa M z_C/t_3)\boldsymbol{h}_{true}$ (an approximation) is used to replace $\hat{\boldsymbol{h}}$ (the optimal solution) to calculate the 2D error, the obtained results is the upper envelope of the 2D error (in millimeters).

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

$$\varepsilon_M^2=\kappa^2\frac{R^6}{36\delta_a^2} \quad \text{(in pixels)}. \tag{43}$$

Many techniques can be used to determine the value of κ, but we recommended to use the method we proposed in Shih,[2] since least efforts is needed to adapt the method described in Section 3 to estimate κ.

4.4. *The envelope 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 by combining (29) and (43):

$$\varepsilon_{\text{Envelope}}^2 \approx \varepsilon_M^2+\varepsilon_n^2. \tag{44}$$

Notice that the second term, ε_n^2, of equation (44) is an expectation value, which means that the violation of the approximate upper envelope, $\varepsilon_{\text{Envelope}}$, is possible.

5. 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, it is easier to control the 3D measurement noise such that it has much smaller effect than the 2D observation noise has. Let σ 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 Weng[4]). The images are of 480 × 512 pixels. The synthetic camera is assumed to have the effective focal length of $f = 25.2847$ mm and the pixel size of $\delta_u = 0.01566$ mm and $\delta_v = 0.013$ mm in horizontal and vertical directions, respectively. The radial lens distortion coefficient is 0.00035 mm^{-2}. The extrinsic camera parameters include three Euler angles, 45.22, 0.95 and 45.52 all in degrees, rotating about z-, y-, z- axis successively, and the transition vector (138.82, 136.81, 1811.11) mm. The calibration and test points are selected from a volume having the depth (in the direction of the optical axis) of 500 mm.

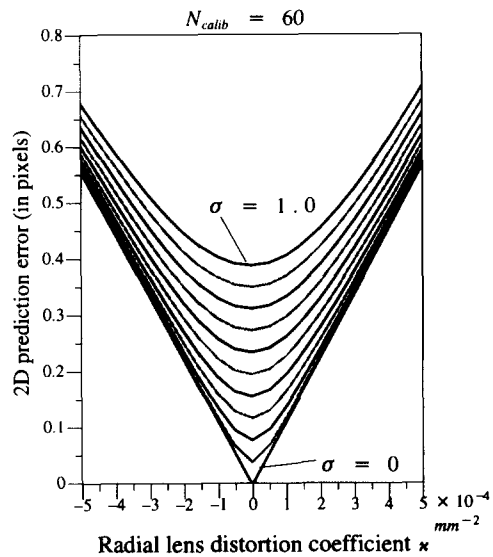

Fig. 5. The simulated 2D prediction error.

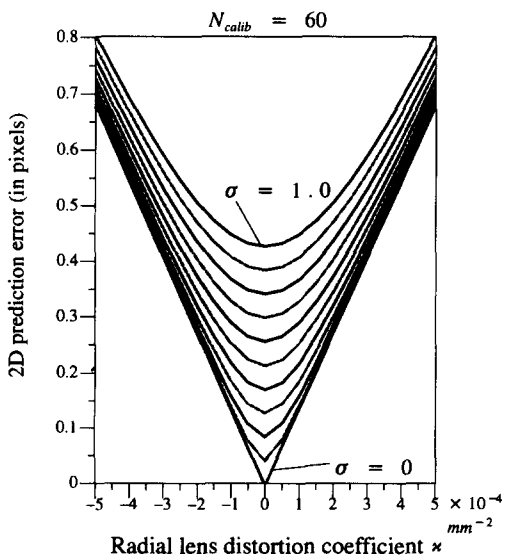

Fig. 6. The predicted error envelope.

The first experiment observed 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 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. $\varepsilon_{\text{Exp}} \propto \kappa$, as we expected, where ε_{Exp} denotes the 2D prediction error evaluated in experiments. Also, it increases as the 2D observation noise or σ increases, i.e. $\varepsilon_{\text{Exp}} \propto \sigma$. Each curve shown in Fig. 5, from bottom to top is obtained by using the 2D observation noise having the standard deviation, $\sigma = 0.0, 0.1, 0.2, \ldots, 1.0$ pixels, respectively. Figure 6 shows the error envelope obtained by using equation (44). Although the basic Assumption 3 does not hold (z-components can vary in the range of 1300–1800 mm), our error envelope still predicts the actual 2D prediction error quite precisely.

This envelope was tested further by the next experiment. Here, four of the intrinsic parameters were generated randomly (see Table 1). The calibration and test points were generated from 20 planes, which were equally spaced with Z_{inc} mm. Hence, we were using a working volume having the depth of 20 × Z_{inc} mm. On each plane, we generated N_p random points for calibration, which yielded totally $N_{\text{calib}} = 20 \times N_p$ calibration points. The reason we set up such a configuration is to simulate the real equipment we have. Both Z_{inc} and N_p in this experiment were also randomly selected (as shown in Table 1), but the number of test points is fixed to 200.

Totally, 10,000 trials were simulated. For each random trial, the computed 2D prediction error was normalized by its theoretic envelope. Figure 7 shows the histogram of the normalized error which shows that, in most trials the 2D prediction error is close to and less than the theoretic envelope, i.e. the normalized error $\lesssim 1$ with high probability. Still, there are some points which exceed the theoretic envelop. This is partially because of that ε_n^2 is an expectation value, not an upper bound. Figures 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 predic-

Table 1. The interval of parameters tested in the second experiment

Parameters	Interval of the uniform distribution
Focal length, f	12.5 mm ~ 75 mm
Principle point, u_0	−20 pixels ~ +20 pixels
Principle point, v_0	−20 pixels ~ +20 pixels
Lens distortion, κ	$-0.0005\,\text{mm}^{-2} \sim +0.0005\,\text{mm}^{-2}$
2D noise, σ	0.0 pixel ~ 1.0 pixel
Distance between successive plane, Z_{inc}	1 mm ~ 25mm
No. of calibration points on each plane, N_p	1 point ~ 10 points*

* Integer random number.

tion error, which are the effective focal length, f, the principle point, (u_0, v_0), and depth of the working volume, see Figs 8–11. In Fig. 14, we can see that when the lens distortion is very small, i.e. $|\kappa| \approx 0$, the effects

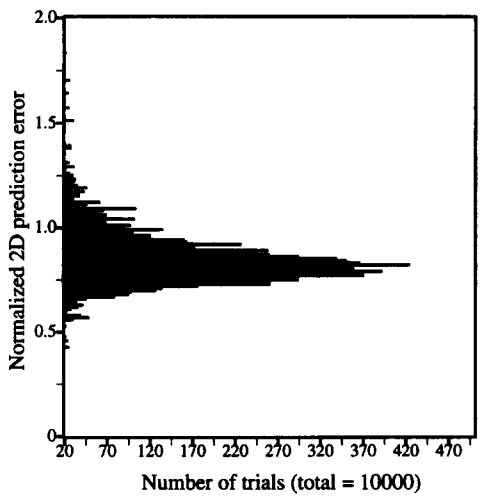

Fig. 7. Histogram of the normalized 2D prediction error with 10,000 trials.

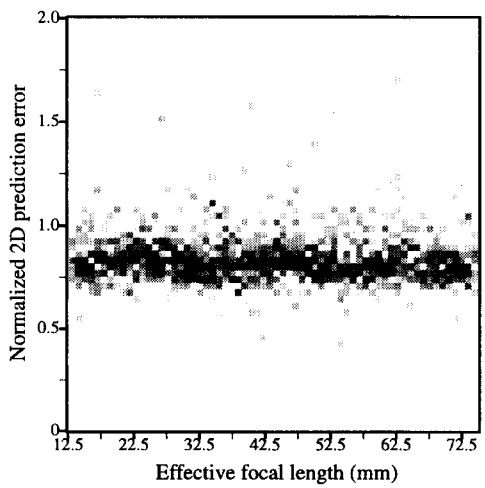

Fig. 8. Distribution of the normalized 2D prediction error with respect to the effective focal length.

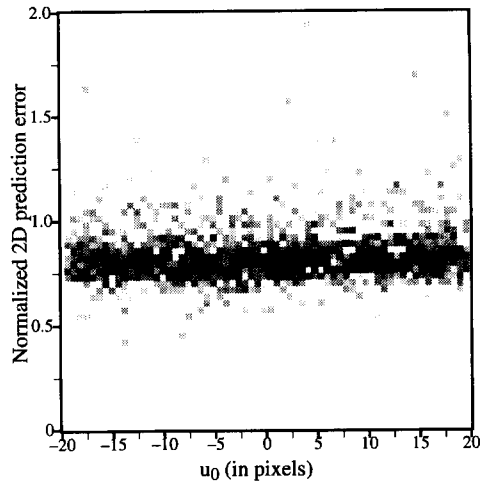

Fig. 9. Distribution of the normalized 2D prediction error with respect to u_0.

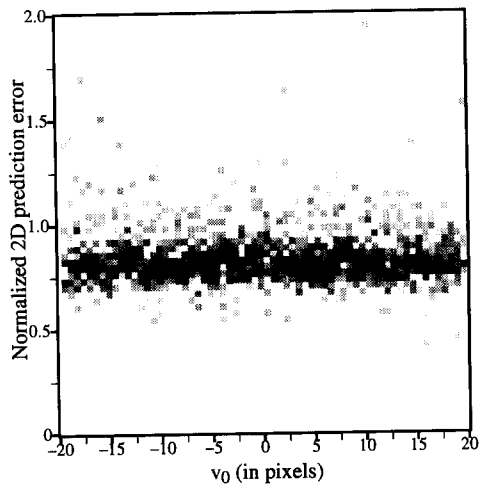

Fig. 10. Distribution of the normalized 2D prediction error with respect to v_0.

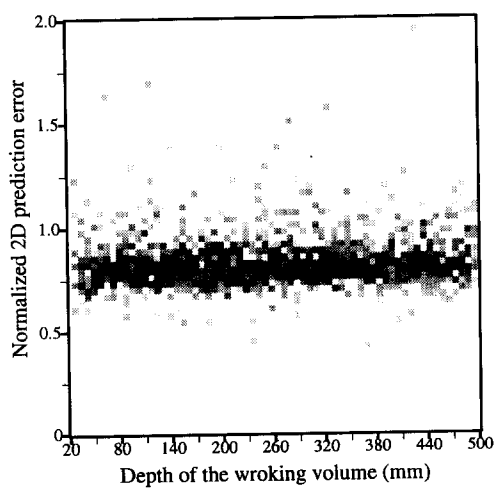

Fig. 11. Distribution of the normalized 2D prediction error with respect to the depth of the working volume.

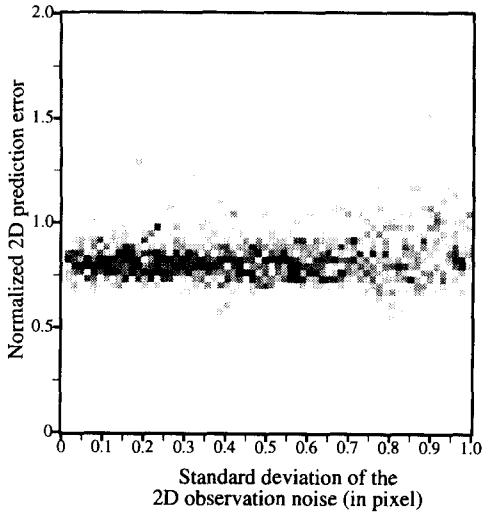

Fig. 12. Distribution of the normalized 2D prediction error with respect to the standard deviation of the 2D observation noise.

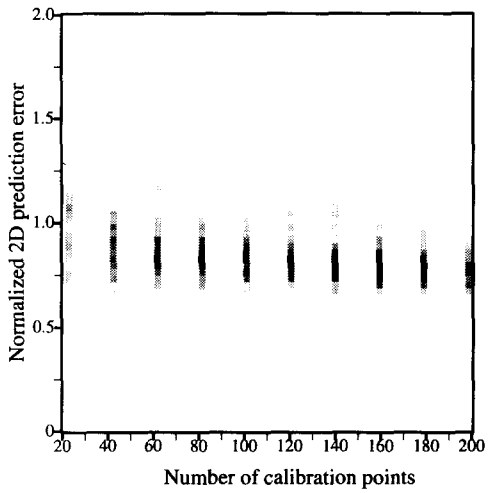

Fig. 13. Distribution of the normalized 2D prediction error with respect to the number of calibration points.

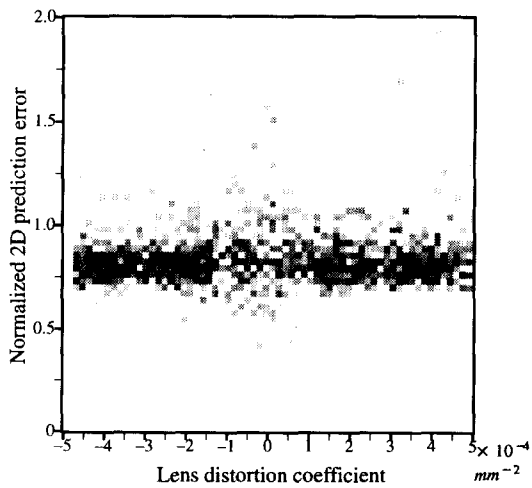

Fig. 14. Distribution of the normalized 2D prediction error with respect to the lens distortion coefficient.

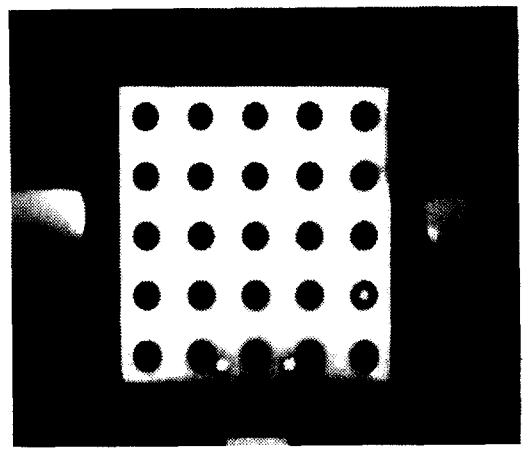

Fig. 15. A typical image of the calibration plate containing 25 calibration points used in the real experiments.

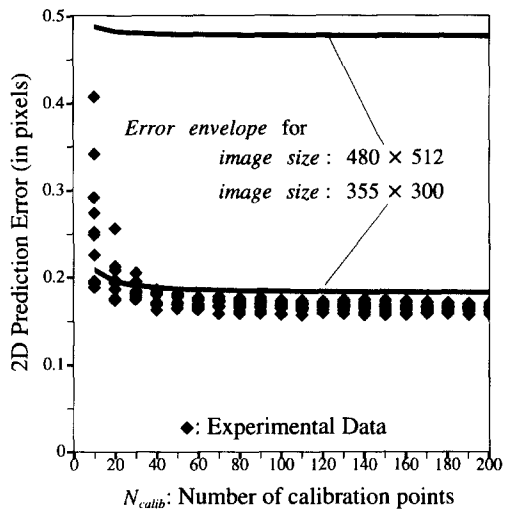

Fig. 16. Comparison of the calibration error obtained in the real experiments with our predicted upper envelope.

of the 2D observation noise dominates, and the $\varepsilon_{\text{Envelope}}$ is more of an error expectation than of an upper envelope. Therefore, the normalized error varies in a larger extent. Figures 12 and 13 show that when the number of calibration points is small or the 2D observation noise is large, the approximate envelope tends to be violated. At the beginning we expect that the smaller the depth of the working volume is (to let the Assumption 3 be true), the more correct the envelope 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 singular the calibration problem tends to be, since the calibration points tends to be on the same plane), we do not see this phenomenon in Fig. 11.

The third experiment tested the envelope 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 was mounted on a translation stage. One image was taken each time the translation stage was moved toward the camera by 25 mm. A typical image is shown in Fig. 15. 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 chose N_{calib} points from the 525 2D–3D pairs to calibrate the camera and used all remaining points to test the calibrated parameters. The above random trials were repeated ten times to obtain ten sets of the 2D prediction error. Figure 16 shows the ten sets of data and two predicted envelopes based on two different effective image sizes (here $\kappa = 0.00035\,\text{mm}^{-2}$, which corresponds to roughly 2–3 pixels of distortion near the four image corners). 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. 15), the envelope calculated with this image size is much closer to the experimental results. To use every pixels in the 480×512 image, the error envelope will be approximately three times of the experimental results.

6. CONCLUSIONS

In this paper, we have derived an approximate upper envelope for the 2D prediction error. The effects of both the radial lens distortion and the 2D observation noise are considered. This envelope was tested by computer simulations and real experiments which show that the upper envelope is quite tight, i.e. it is close to the experimental results and still envelopes 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 Blostein.[12] Thus, the error envelope can be used as a criterion for deciding 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 envelope:

(1) Determine the acceptable 2D prediction error and 3D angular error. If the specified error envelope is given as the 3D angular error, then equation (13) is used to translate it to the 2D prediction error. For convenience, let us denote this specified 2D error envelope as ε_{spec}.

(2) Calculate the approximate error envelope, $\varepsilon_{Envelope}$, by equation (44) according to the parameters of the equipments to be used.

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

(4) If $\varepsilon_{spec} < \varepsilon_{Envelope}$ then try to reduce ε_n in equation (44) as much as possible, by making the feature extraction more accurate (reduce σ) and by increasing the number of calibration points. Check if this process brings the theoretic envelope, $\varepsilon_{Envelope}$, to the value smaller than the specified one, ε_{spec}.

(5) If $\varepsilon_{Envelope}$ still cannot meet the requirement after the reduction of ε_n in step 4), then try to reduce the effective size of the image to an acceptable level [see equation (43)].

(6) If the efforts in steps (4) and (5) fail to reduce $\varepsilon_{Envelope}$ such that $\varepsilon_{spec} > \varepsilon_{Envelope}$, then a nonlinear camera model should be considered in the camera calibration procedure as in Faig,[1] Shih,[2] Tsai[3] and Weng.[4]

A linear camera model is always the first consideration of engineers. Not only will it simplify the camera calibration procedure, but it will 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.

REFERENCES

1. W. Faig, Calibration of close-range photogrammetry systems: mathematical formulation, *Photogrammetric Engineering and Remote Sensing* **41**(12), 1479–1486 (1975).
2. S. W. Shih, Y. P. Hung and W. S. Lin, Accurate linear technique for camera calibration considering lens distortion by solving an eigenvalue problem, *Optical Engineering* **32**(1), 138–149 (1993).
3. 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* **RA-3**(4), 323–344 (1987).
4. J. Weng, P. Cohen and M. Herniou, Camera calibration with distortion models and accuracy evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(10), 965–980 (1992).
5. O. D. Faugeras and G. Toscani, The calibration problem for stereo, *Proceedings Conf. of Computer Vision and Pattern Recognition* 15–20 (1986).
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 (1990).
7. T. M. Start, Recovering the camera parameters from a transformation matrix, DARPA Image Understanding Workshop, 264–271 (1984).
8. I. Sutherland, Three-dimensional data input by tablet, *Proceedings of the IEEE*, **62**(4), 453–461 (1974).
9. S. Ganapaphy, Decomposition of transformation matrices for robot vision, *Proceedings Int. Conf. on Robotics and Automation* 130–139 (1984).
10. Y. P. Hung and S. W. Shih, When should we consider lens distortion in camera calibration, *IAPR Workshop on Machine Vision Applications*, Tokyo, 367–370 (1990).
11. R. O. Duda and P. E. Hart, *Pattern Recognition and Scene Analysis*, Wiley, New York (1973).
12. S. D. Blostein and T. S. Hung, Error analysis in stereo determination of 3-D point positions, *IEEE Trans. Pattern Anal. Machine Intel.* **9**(6), 752–765 (1987).

About the Author—SHENG-WEN SHIH received the M.S. degree from the National Taiwan University in electrical engineering in 1990. Since then he joined the Institute of Information Science, Academia Sinica. He is now also a Ph.D. student in the Institute of Electrical Engineering at the National Taiwan University. His current research interests are in active vision, image sequence analysis and robotics.

About the Author—YI-PING HUNG received the B.S.E.E. degree from the National Taiwan University in 1982, the M.Sc. degree in electrical engineering, the M.Sc. degree in applied mathematics, the Ph.D. degree in electrical engineering, all from Brown University, in 1987, 1988 and 1990, respectively. He is currently an associate research fellow in the Institute of Information Science, Academia Sinica, and an adjunct associate professor at the National Taiwan University. His research interests include computer vision and robotics.

About the Author—WEI-SONG LIN was born in Taiwan R.O.C. in 1951. He received the B.S. degree in engineering science and M.S. degree in electrical engineering from National Cheng Kung University, Taiwan, R.O.C. in 1973 and 1975, respectively, and attained a distinguished paper award from the Association of Chinese Electrical Engineers at this time. He received the Ph.D. degree from the Institute of Electrical Engineering of National Taiwan University in 1982 and where he became an Associate Professor during 1983–1987. From 1982 to 1984 he was also the head of Electronic Instrument Division of Ching Ling Industrial Research Center where he worked mainly on design and implementation of electronic instruments. Since 1987 he has been a Professor at National Taiwan University. He is now a member of the International Association of Science and Technology for Development (IASTED), and National Committee of the Internal Union of Radio Science (URSI). He has received excellent research award of the National Science Council in 1991. His current research interests are in the field of computer control, computer-based sensing and instrumentation, analysis and design of dynamic control systems, system engineering and automation.

APPENDIX A

Rewrite equations (6a) and (6b) as

$$(1-\kappa\varrho^2)(u-u_0)[x\ y\ z\ 1]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}=[x\ y\ z\ 1]\begin{bmatrix}r_1f/\delta_u\\r_2f/\delta_u\\r_3f/\delta_u\\t_1f/\delta_u\end{bmatrix} \tag{A1}$$

$$(1-\kappa\varrho^2)(v-v_0)[x\ y\ z\ 1]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}=[x\ y\ z\ 1]\begin{bmatrix}r_4f/\delta_v\\r_5f/\delta_v\\r_6f/\delta_v\\t_2f/\delta_v\end{bmatrix} \tag{A2}$$

From equation (A1) we have

$$(u-u_0)[x\ y\ z\ 1]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}-\kappa\varrho^2(u-u_0)[x\ y\ z\ 1]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}=[x\ y\ z\ 1]\begin{bmatrix}r_1f/\delta_u\\r_2f/\delta_u\\r_3f/\delta_u\\t_1f/\delta_u\end{bmatrix} \tag{A3}$$

which leads to

$$[x\ y\ z\ 1]\begin{bmatrix}r_1f/\delta_u+r_7u_0\\r_2f/\delta_u+r_8u_0\\r_3f/\delta_u+r_9u_0\\t_1f/\delta_u+t_3u_0\end{bmatrix}+[-ux-uy-uz-u]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}+\kappa\varrho^2(u-u_0)[x\ y\ z\ 1]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}=0. \tag{A4}$$

Similarly, from (A2) we have

$$[x\ y\ z\ 1]\begin{bmatrix}r_4f/\delta_v+r_7v_0\\r_5f/\delta_v+r_8v_0\\r_6f/\delta_v+r_9v_0\\t_2f/\delta_v+t_3v_0\end{bmatrix}+[-vx-vy-vz-v]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}+\kappa\varrho^2(v+v_0)[x\ y\ z\ 1]\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix}=0. \tag{A5}$$

For convenience, let us define the coefficient matrix, $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, for camera calibration and some composite parameters, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, $\boldsymbol{p}$ and $\boldsymbol{q}$, as following:

$$\boldsymbol{A}\equiv\begin{bmatrix}\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\x_j&y_j&z_j&1&0&0&0&0\\0&0&0&0&x_j&y_j&z_j&1\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\end{bmatrix},$$

$$\boldsymbol{B}\equiv\begin{bmatrix}\vdots&\vdots&\vdots&\vdots\\-u_jx_j&-u_jy_j&-u_jz_j&-u_j\\-v_jx_j&-v_jy_j&-v_jz_j&-v_j\\\vdots&\vdots&\vdots&\vdots\end{bmatrix}$$

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

$$\boldsymbol{P}_1\equiv\begin{bmatrix}r_1f/\delta_u+r_7u_0\\r_2f/\delta_u+r_8u_0\\r_3f/\delta_u+r_9u_0\\t_1f/\delta_u+t_3u_0\end{bmatrix},\quad\boldsymbol{P}_2\equiv\begin{bmatrix}r_4f/\delta_v+r_7v_0\\r_5f/\delta_v+r_8v_0\\r_6f/\delta_v+r_9v_0\\t_2f/\delta_v+t_3v_0\end{bmatrix},\quad\boldsymbol{P}_3\equiv\begin{bmatrix}r_7\\r_8\\r_9\\t_3\end{bmatrix},$$

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

With the definitions of $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$, and applying (A4) and (A5) for all 2D–3D pairs, we have

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \end{bmatrix} + \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_j x_j & -u_j y_j & -u_j z_j & -u_j \\ -v_j x_j & -v_j y_j & -v_j z_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \boldsymbol{P}_3$$

$$+ \kappa \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ (u_j - u_0)\varrho_j^2 x_j & (u_j - u_0)\varrho_j^2 y_j & (u_j - u_0)\varrho_j^2 z_j & (u_j - u_0)\varrho_j^2 \\ (v_j - v_0)\varrho_j^2 x_j & (v_j - v_0)\varrho_j^2 y_j & (v_j - v_0)\varrho_j^2 z_j & (v_j - v_0)\varrho_j^2 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \boldsymbol{P}_3 = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{A6}$$

with

$$\varrho_j^2 = \delta_u^2 (u_j - u_0)^2 + \delta_v^2 (v_j - v_0)^2.$$

Dividing equation (A6) by t_3 and substituting the definitions of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{p}$ and $\boldsymbol{q}$ into it, we have

$$\boldsymbol{A}\boldsymbol{p} + \boldsymbol{B}\boldsymbol{q} + \kappa \boldsymbol{C}\boldsymbol{q} = 0. \tag{A7}$$

APPENDIX B

As shown in Fig. B1(a) and (b), the average distance from the image point to the estimated lens center, d_a, and the average pixel spacing, δ_a, can be computed through the following equations:

$$d_a \equiv \frac{1}{2\pi\eta_r} \int_0^{\eta_r} \int_0^{2\pi} r(\eta)\, d\gamma\, d\eta = \frac{f}{\eta_r} \ln|\sec(\eta_r) + \tan(\eta_r)|, \tag{B1}$$

$$\delta_a \equiv \frac{2}{\pi} \int_0^{\pi/2} \delta(\alpha)\, d\alpha = \frac{2}{\pi} [\delta_u \ln|\sec(\alpha_u) + \tan(\alpha_u)| + \delta_v \ln|\sec(\alpha_v) + \tan(\alpha_v)|], \tag{B2}$$

where

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

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

$$\alpha_u \equiv \tan^{-1}\left(\frac{\delta_v}{\delta_u}\right), \quad \text{and} \quad \alpha_v \equiv \tan^{-1}\left(\frac{\delta_u}{\delta_v}\right).$$

APPENDIX C

We first show that $z_{Cj}/t_3 = h_9 x_j + h_{10} y_j + h_{11} z_j + 1$. By direct computation, since

$$z_{Cj} = [x_j \; y_j \; z_j \; 1]\boldsymbol{P}_3 = [x_j \; y_j \; z_j \; 1] \begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix}$$

$$= [x_j \; y_j \; z_j \; 1]\boldsymbol{q} \times t_3$$

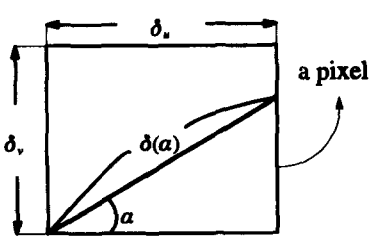

Fig. B1(a). The average pixel spacing.

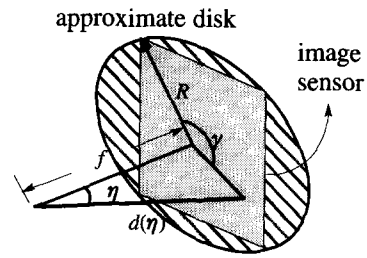

Fig. B1(b). The average distance to the estimated lens center.

and $\boldsymbol{q} = [h_9 \; h_{10} \; h_{11} \; 1]$, it is easy to see that

$$\frac{z_{Cj}}{t_3} = h_9 x_j + h_{10} y_j + h_{11} z_j + 1, \tag{C1}$$

and thus

$$\frac{\hat{z}_{Cj}}{\hat{t}_3} = \hat{h}_9 x_j + \hat{h}_{10} y_j + \hat{h}_{11} z_j + 1, \tag{C2}$$

Now we shall prove that

$$\|\Lambda \boldsymbol{h} - \boldsymbol{b}\|^2 = \sum_{j=1}^{N_{\text{calib}}} \left[(u_j - \hat{u}_j)^2 \left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right)^2 + (v_j - \hat{v}_j)^2 \left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right)^2 \right]. \tag{C3}$$

By the definition of $\hat{u}$ and $\hat{z}_C/\hat{t}_3$, we have

$$(\hat{u}_j - u_j)\left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right) = \left(\frac{\hat{h}_1 x_j + \hat{h}_2 y_j + \hat{h}_3 z_j + \hat{h}_4}{\hat{h}_9 x_j + \hat{h}_{10} y_j + \hat{h}_{11} z_j + 1} - u_j\right) \times (\hat{h}_9 x_j + \hat{h}_{10} y_j + \hat{h}_{11} z_j + 1)$$

$$= [x_j \; y_j \; z_j \; 1 \; 0 \; 0 \; 0 \; 0 \; -u_j x_j \; -u_j y_j \; -u_j z_j] \times \boldsymbol{h} - u_j$$

$$= \Lambda_{2j} \boldsymbol{h} - \boldsymbol{b}_{2j}, \tag{C4}$$

where Λ_{2j} and $\boldsymbol{b}_{2j}$ denote the $2j$th row of Λ and $\boldsymbol{b}$, respectively. Similarly, for the v-direction, we have

$$(\hat{v}_j - v_j)\left(\frac{\hat{z}_{Cj}}{\hat{t}_3}\right) = [0 \; 0 \; 0 \; 0 \; x_j \; y_j \; z_j \; 1 \; -v_j x_j \; -v_j y_j \; -v_j z_j] \times \boldsymbol{h} - v_j$$

$$= \Lambda_{2j+1} \boldsymbol{h} - \boldsymbol{b}_{2j+1}. \tag{C5}$$

From (C4) and (C5), it is obvious that (C3) holds.

APPENDIX D

Substituting equation (12) into (26), we have

$$E[\|\boldsymbol{b}_T - \Lambda \hat{\boldsymbol{h}}\|^2] = E\{\|\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t \boldsymbol{e}\|^2\} = E\{\boldsymbol{e}^t \Lambda (\Lambda^t\Lambda)^{-1}\Lambda^t \boldsymbol{e}\}. \tag{D1}$$

Since equation (59) is a scalar equation, taking trace operation of its both sides will not affect the results. Therefore, we have

$$E\{\boldsymbol{e}^t\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\boldsymbol{e}\} = E\{\text{trace}[\boldsymbol{e}^t\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\boldsymbol{e}]\}, \quad \text{(D2)}$$

which yields

$$E\{\text{trace}[\boldsymbol{e}^t\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\boldsymbol{e}^t]\} = E\{\text{trace}[\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t(\boldsymbol{e}\,\boldsymbol{e}^t)]\}. \quad \text{(D3)}$$

From (D3), we have

$$E\{\text{trace}[\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t(\boldsymbol{e}\,\boldsymbol{e}^t)]\} = \text{trace}\{\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\cdot E[\boldsymbol{e}\,\boldsymbol{e}^t]\}. \quad \text{(D4)}$$

According to the i.i.d. assumption (see Assumption 2), we have that the covariance matrix, $E[\boldsymbol{e}\,\boldsymbol{e}^t]$, is diagonal with the same diagonal element, i.e.

$$E[\boldsymbol{e}\,\boldsymbol{e}^t] = \frac{\sigma^2 z_C^2}{t_3^2}\boldsymbol{I} \quad \text{(D5)}$$

Substituting (D5) into (D4), we have

$$\text{trace}\{\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\cdot E[\boldsymbol{e}\,\boldsymbol{e}^t]\} = \frac{\sigma^2 z_C^2}{t_3^2}\,\text{trace}\{\Lambda(\Lambda^t\Lambda)^{-1}\Lambda^t\},$$

which leads to

$$\frac{\sigma^2 z_C^2}{t_3^2}\,\text{trace}\{(\Lambda^t\Lambda)^{-1}\Lambda^t\Lambda\} = \frac{\sigma^2 z_C^2}{t_3^2}\,\text{trace}\{\boldsymbol{I}_{11\times 11}\}$$

Therefore, we have

$$E[\|\boldsymbol{b}_T - \Lambda\hat{\boldsymbol{h}}\|^2] = \frac{11\sigma^2 z_C^2}{t_3^2}. \quad \text{(D6)}$$

APPENDIX E

From equation (14), we have

$$\hat{u} = \frac{\hat{h}_1 x + \hat{h}_2 y + \hat{h}_3 z + \hat{h}_4}{\hat{h}_9 x + \hat{h}_{10} y + \hat{h}_{11} z + 1} \times \frac{1 - \kappa M z_C/t_3}{1 - \kappa M z_C/t_3}.$$

Recall that from (33)

$$\boldsymbol{h}_{\text{true}} \approx (1 - \kappa M z_C/t_3)\hat{\boldsymbol{h}}, \quad \text{(E1)}$$

i.e.

$$h_i \approx \hat{h}_i \times (1 - \kappa M z_C/t_3), \quad i = 1, 2, \ldots, 11,$$

we have

$$\hat{u} \approx \frac{h_1 x + h_2 y + h_3 z + h_4}{h_9 x + h_{10} y + h_{11} z + 1 - \kappa M z_C/t_3} \quad \text{(E2)}$$

Since $z_C = (h_9 x + h_{10} y + h_{11} z + 1) \times t_3$ (see Appendix 3), equation (E2) can be rewritten as

$$= \frac{h_1 x + h_2 y + h_3 z + h_4}{h_9 x + h_2 y + h_3 z + 1} \times \frac{1}{1 - \kappa M}. \quad \text{(E3)}$$

In practice $\kappa M \approx \kappa R_F^2 \ll 1$, thus

$$\hat{u} \approx u_{ud} \times (1 + \kappa M). \quad \text{(E4)}$$

Similarly, it can be shown that

$$\hat{v} \approx v_{ud} \times (1 + \kappa M). \quad \text{(E5)}$$

Since $\kappa M \approx \kappa R_F^2 \ll 1$, and the principle point (u_0, v_0) is small comparing to most of the image points, so that $(\kappa M u_0, \kappa M v_0)$ is negligible, which yields [from (E4) and (E5)]

$$\begin{bmatrix} \hat{u} - u_0 \\ \hat{v} - v_0 \end{bmatrix} \approx (1 + \kappa M)\begin{bmatrix} u_{ud} - u_0 \\ v_{ud} - v_0 \end{bmatrix}. \quad \text{(E6)}$$