TR-91-018

An Efficient and Accurate Camera Calibration Technique for 3D Computer Vision

Sheng-Wen Shih,
Yi-Ping Hung, and
Wei-Song Lin

AN EFFICIENT AND ACCURATE CAMERA CALIBRATION TECHNIQUE FOR 3D COMPUTER VISION

Sheng-Wen Shih, Yi-Ping Hung
Institute of Information Science
Academia Sinica

Wei-Song Lin
Institute of Electrical Engineering
National Taiwan University

ABSTRACT

In this report we propose a new technique for calibrating a camera with very high accuracy and low computational cost. The geometric camera parameters considered include camera position, orientation, focal length, radial lens distortion, pixel size and optical axis piercing point. With our method, the camera parameters to be estimated are divided into two parts: the radial lens distortion coefficient κ, and a composite parameter vector **c** composed of all the above geometric camera parameters other than κ. Instead of using nonlinear optimization techniques, the estimation of κ is transformed into an eigenvalue problem of a 8×8 matrix. Our method is fast since it requires only linear computation. It is accurate since the effect of the lens distortion is considered and all the information contained in the calibration points is used. Computer simulation and real experiment have shown that the performance of our calibration method is better than that of the well-known method proposed by Tsai[8].

I. INTRODUCTION

I.1 The Importance of Camera Calibration

Camera calibration in the context of three-dimensional machine vision, as defined in Tsai[8], is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters). 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.

For the applications that need the above two transformations, e.g., automatic assembling, gauging, tracking, robot calibration, trajectory analysis and vehicle guidance, it is essential to calibrate the camera on-line or off-line. In general, the system performance is dominated by the accuracy of camera calibration. The trade-off between the accuracy and the cost mainly depends on the requirements of the application, and in general, the accuracy is proportional to the cost.

I.2 Existing Techniques for Camera Calibration

Many techniques have been developed for camera calibration because of the strong demand of applications. These techniques can be classified into two categories: one that considers lens distortion [2] [8] [9] [10], and one that neglects lens distortion [3] [6] [7] [13]. A typical linear technique that does not consider lens distortion is the one estimating the perspective transformation matrix **H** [7]. 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 [4][5][6].

Faig's method [2] is a good example of 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 the camera parameters with an objective to minimize residual errors of these equations. One disadvantage of this kind of method is that a good initial guess is required to start the nonlinear search and it is also computationally expensive.

A few years ago, Tsai proposed an efficient two stage technique using the "radial alignment constraint"[8]. 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 [9]. Our experience also shows [12] that Tsai's method can be worse than the simple linear method of [7] if lens distortion is relatively small.

Recently, Weng presents some experimental results using a two–step method[9]. 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.

In this report we develop a method, which not only has the advantage of the linear calibration method but also maintains the accuracy of those considering lens distortion. Section II introduces the camera model adopted in this report. Section III describes the new calibration technique. Section IV presents the results of computer simulation and real experiment, which show the performance of this method is better than Tsai's method.

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 mini-meters

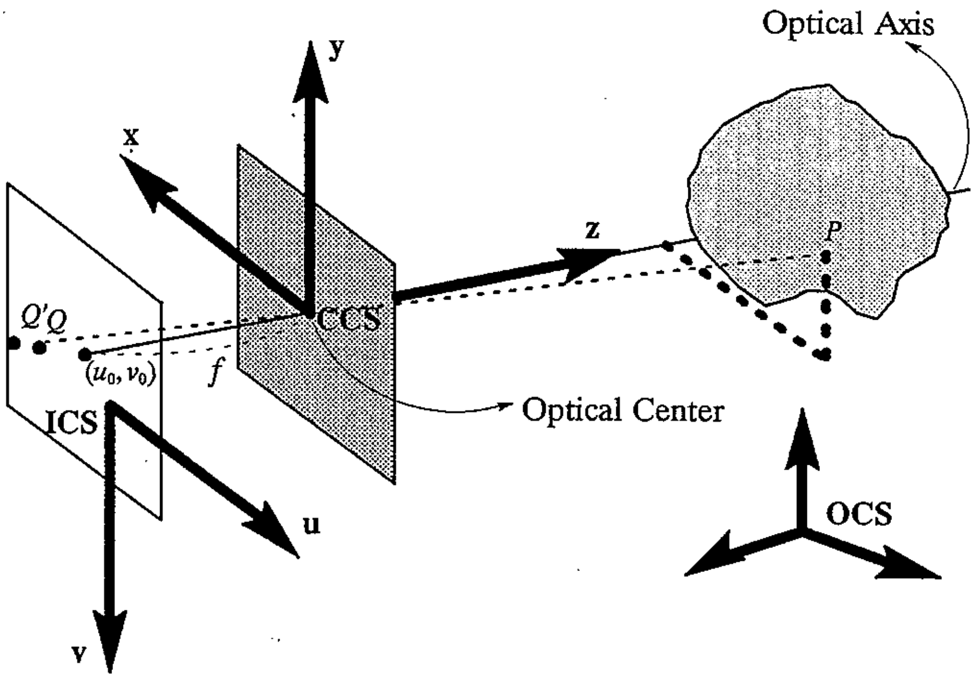
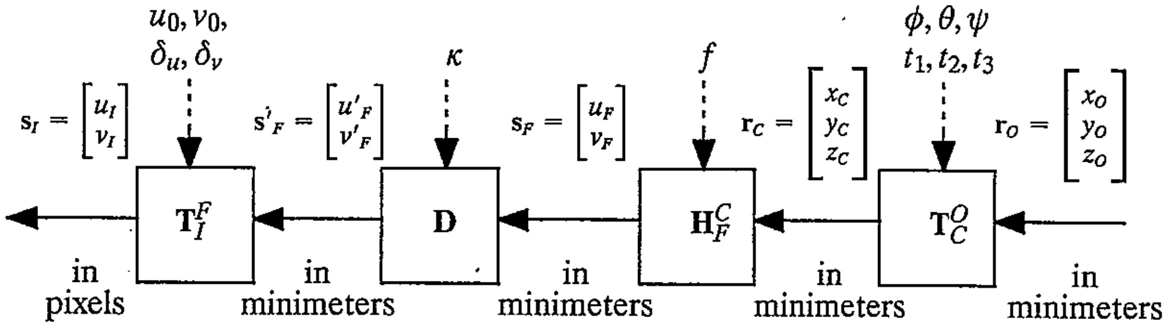


Fig. 1 Pinhole camera model with lens distortion, where P is a 3D object point, Q and Q' are its undistorted and distorted image points, respectively.
OCS -- Object Coordinate System (3D)
CCS -- Camera Coordinate System (3D)
ICS -- computer Image Coordinate System (2D)

(mm), with respect to a fixed object coordinate system (OCS). Let the camera coordinate system (CCS), also in mm, 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 optical 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 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:

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



$u_0, v_0, \delta_u, \delta_v$
κ
f
ϕ, θ, ψ
t_1, t_2, t_3
$\mathbf{s}_I = \begin{bmatrix} u_I \\ v_I \end{bmatrix}$
$\mathbf{s}'_F = \begin{bmatrix} u'_F \\ v'_F \end{bmatrix}$
$\mathbf{s}_F = \begin{bmatrix} u_F \\ v_F \end{bmatrix}$
$\mathbf{r}_C = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix}$
$\mathbf{r}_O = \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix}$
$\mathbf{T}_I^F$
$\mathbf{D}$
$\mathbf{H}_F^C$
$\mathbf{T}_C^O$
in pixels
in minimeters
in minimeters
in minimeters
in minimeters

Fig. 2 Relation between different transformation matrices

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

$$\begin{cases} x_C = r_1\, x_O + r_2\, y_O + r_3\, z_O + t_1 \\ y_C = r_4\, x_O + r_5\, y_O + r_6\, z_O + t_2 \\ z_C = r_7\, x_O + r_8\, y_O + r_9\, z_O + t_3 \end{cases} \tag{1.1}$$

or

$$\tilde{\mathbf{r}}_C = \mathbf{T}_C^O\, \tilde{\mathbf{r}}_O \qquad \textit{with} \qquad \mathbf{T}_C^O = \begin{bmatrix} \mathbf{R}_C^O & \mathbf{t}_C^O \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

where tilde ($\tilde{}$) denotes homogeneous coordinates [1], $\mathbf{t}_C^O = (t_1 \;\; t_2 \;\; t_3)^t$ is a translation vector, and $\mathbf{R}_C^O$ is a 3x3 rotation matrix determined by the three Euler angles, ϕ, θ, ψ, 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 mm) of the undistorted image point Q lying on the image plane. Then, we have

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

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

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

II.3 Lens distortion from Q to Q'

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

$$\begin{cases} u_F = (1-\kappa\,\varrho^2)\;u'_F \\ v_F = (1-\kappa\,\varrho^2)\;v'_F \end{cases} \quad \textit{where} \quad \varrho^2 = u'^2_F + v'^2_F \tag{3.1}$$

or

$$\mathbf{s}_F = (1-\kappa\;||\,\mathbf{s}'_F||^{\,2})\;\mathbf{s}'_F \tag{3.2}$$

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

II.4 Scaling and translation of 2D image coordinates

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

$$\begin{cases} u'_F = (u_I - u_0)\;\delta_u \\ v'_F = (v_I - v_0)\;\delta_v \end{cases} \tag{4.1}$$

or

$$\tilde{\mathbf{s}}_I = \mathbf{T}_I^F\;\tilde{\mathbf{s}}'_F \qquad \textit{with} \qquad \mathbf{T}_I^F = \begin{bmatrix} 1/\delta_u & 0 & u_0 \\ 0 & 1/\delta_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

where δ_u and δ_v are the horizontal and vertical pixel spacing (mm/pixel), u_0 and v_0 are the coordinates (in pixels) of the origin of the CCS in the computer image coordinate system.

Using the above notations for camera parameters, the geometric camera parameters β $= [t_1 \ t_2 \ t_3 \ \phi \ \theta \ \psi \ \ f \ \kappa \ \delta_u \ u_0 \ v_0]^t$. 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 (1.1), (2.1), (3.1) and (4.1), 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 $$\varrho = \sqrt{\delta_u^2(u_I-u_0)^2 + \delta_v^2(v_I-v_0)^2}$$

III. THE NEW CAMERA CALIBRATION TECHNIQUE

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 coefficient κ and the composite parameters **c** (as described following equation(10)), then decompose them into β [4][5][6]. Two similar calibration algorithms are described in this report; one requires a set of *noncoplanar* calibration points, and the other only needs *coplanar* calibration points. While the former require noncoplanar calibration points, it is more accurate than the latter.

Our method needs initial guess for u_0, v_0, and δ_u, which can be easily set as follows. Let f_{camera} denote the pixel scanning rate of the camera (e.g., $f_{camera} = 14.31818$ MHz for PULNiX TM–745E), and $f_{digitizer}$ denote the pixel scanning rate of the digitizer or frame grabber (e.g., $f_{digitizer} = 10$ MHz for ITI Series 151). Let δ_u' denote the horizontal pixel spacing of the solid state imager (e.g., $\delta_u' = 11 \ \mu m$ for PULNiX TM–745E). Then a good estimate of δ_u can be obtained by the following equation

$$\hat{\delta}_u = \delta_u' \frac{f_{camera}}{f_{digitizer}}. \tag{6}$$

The other two parameters $(\hat{u}_0, \hat{v}_0)$ can be temporally set to (0, 0), refer to [8], if no other reliable informations about them are available. More accurate estimates of the three parameters can be obtained with our calibration procedure, which can then be used as a better initial guess iteratively. If the amount of radial lens distortion is small, say $|\kappa| \leq 0.0008 \text{ mm}^{-2}$, and the computational speed is not of the main concern, then another two or three iterations can give even higher accuracy according to our experience (see Fig. 9).

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.1 The new camera calibration technique using noncoplanar calibration points

Rewrite equation (5.1) and (5.2) as

$$(1-\kappa\varrho^2)(u-u_0)\begin{bmatrix} x & y & z & 1 \end{bmatrix}\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}\begin{bmatrix} r_1f/\delta_u \\ r_2f/\delta_u \\ r_3f/\delta_u \\ t_1f/\delta_u \end{bmatrix} \tag{7.1}$$

$$(1-\kappa\varrho^2)(v-v_0)\begin{bmatrix} x & y & z & 1 \end{bmatrix}\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}\begin{bmatrix} r_4f/\delta_v \\ r_5f/\delta_v \\ r_6f/\delta_v \\ t_2f/\delta_v \end{bmatrix} \tag{7.2}$$

From equation (7.1) we have

$$(u-u_0)\begin{bmatrix} x & y & z & 1 \end{bmatrix}\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} - \kappa\varrho^2(u-u_0)\begin{bmatrix} x & y & z & 1 \end{bmatrix}\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}\begin{bmatrix} r_1f/\delta_u \\ r_2f/\delta_u \\ r_3f/\delta_u \\ t_1f/\delta_u \end{bmatrix} \tag{8}$$

which leads to

$$[x \ \ y \ \ z \ \ 1]\begin{bmatrix} r_1 f/\delta_u + r_7 u_0 \\ r_2 f/\delta_u + r_8 u_0 \\ r_3 f/\delta_u + r_9 u_0 \\ t_1 f/\delta_u + t_3 u_0 \end{bmatrix} + [-ux \ \ -uy \ \ -uz \ \ -u]\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} + \kappa\varrho^2(u-u_0)[x \ \ y \ \ z \ \ 1]\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} = 0\,. \quad (9)$$

Similarly, from (7.2) we have

$$[x \ \ y \ \ z \ \ 1]\begin{bmatrix} r_4 f/\delta_v + r_7 v_0 \\ r_5 f/\delta_v + r_8 v_0 \\ r_6 f/\delta_v + r_9 v_0 \\ t_2 f/\delta_v + t_3 v_0 \end{bmatrix} + [-vx \ \ -vy \ \ -vz \ \ -v]\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} + \kappa\varrho^2(v-v_0)[x \ \ y \ \ z \ \ 1]\begin{bmatrix} r_7 \\ r_8 \\ r_9 \\ t_3 \end{bmatrix} = 0\,. \quad (10)$$

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

Then using (9) and (10), for all 2D–3D pairs, we have

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

$$+\,\kappa\begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ (u_j-u_0)\varrho_j^2 x_j & (u_j-u_0)\varrho_j^2 y_j & (u_j-u_0)\varrho_j^2 z_j & (u_j-u_0)\varrho_j^2 \\ (v_j-v_0)\varrho_j^2 x_j & (v_j-v_0)\varrho_j^2 y_j & (v_j-v_0)\varrho_j^2 z_j & (v_j-v_0)\varrho_j^2 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}\mathbf{P}_3 = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (11)$$

with

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

Define

$$\mathbf{A} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \qquad \mathbf{B} \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -u_jx_j & -u_jy_j & -u_jz_j & -u_j \\ -v_jx_j & -v_jy_j & -v_jz_j & -v_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

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

$$\mathbf{p} \equiv \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix} / \parallel \mathbf{P}_3 \parallel \quad \text{and} \quad \mathbf{q} \equiv \mathbf{P}_3 / \parallel \mathbf{P}_3 \parallel . \tag{12}$$

In practice, the image center (u_0, v_0) is relatively small comparing to almost all the image feature points (u_j, v_j), and the estimate of δ_u obtained by equation (6) is usually quite accurate. Therefore, in the following computation we can use some initial estimates of u_0, v_0 and δ_u in the matrix C as if they were the true values. The perturbation of the matrix C caused by the estimation error in $\hat{u}_0$, $\hat{v}_0$ and $\hat{\delta}_u$ is negligible in general, since κ is usually very small and the perturbation of the matrix C times κ in equation (11) is trivial. In other words, the less the amount of the radial lens distortion is, the better the performance of the proposed technique will have.

Since the 2D observation noise always exists, equation (11) will not be exactly zero, i.e.,

$$\mathbf{A}\mathbf{p} + \mathbf{B}\mathbf{q} + \kappa\mathbf{C}\mathbf{q} = \epsilon \neq 0 . \tag{13}$$

Hence, the parameters to be estimated, κ and $\mathbf{c}$ (or equivalently, κ, $\mathbf{p}$, and $\mathbf{q}$) can be computed by minimizing the following criterion E with respect to κ, $\mathbf{p}$, and $\mathbf{q}$:

$$E \equiv \parallel \mathbf{A}\mathbf{p} + \mathbf{B}\mathbf{q} + \kappa\mathbf{C}\mathbf{q} \parallel^2 ,$$

subject to the constraint: $\parallel \mathbf{q} \parallel^2 = 1$.

To minimize E, we form the Lagrangian

$$L \equiv \| \mathbf{Ap} + \mathbf{Bq} + \kappa\mathbf{Cq} \|^2 + \lambda(1 - \| \mathbf{q} \|^2)\,, \tag{14}$$

and set $\partial L/\partial \mathbf{p} = 0$, $\partial L/\partial \mathbf{q} = 0$, $\partial L/\partial \lambda = 0$ and $\partial L/\partial \kappa = 0$, which yields

$$\frac{1}{2}\frac{\partial L}{\partial \mathbf{p}} = \mathbf{A}^t\mathbf{Ap} + \mathbf{A}^t\mathbf{Bq} + \kappa\mathbf{A}^t\mathbf{Cq} = 0 \tag{15}$$

$$\frac{1}{2}\frac{\partial L}{\partial \mathbf{q}} = \mathbf{B}^t\mathbf{Bq} + \kappa^2\mathbf{C}^t\mathbf{Cq} + \mathbf{B}^t\mathbf{Ap} + \kappa\mathbf{C}^t\mathbf{Ap} + \kappa\mathbf{C}^t\mathbf{Bq} + \kappa\mathbf{B}^t\mathbf{Cq} - \lambda\mathbf{q} = 0\,. \tag{16}$$

$$\frac{1}{2}\frac{\partial L}{\partial \kappa} = \mathbf{q}^t\mathbf{C}^t\mathbf{Ap} + \mathbf{q}^t\mathbf{C}^t\mathbf{Bq} + \kappa\mathbf{q}^t\mathbf{C}^t\mathbf{Cq} = 0\,. \tag{17}$$

From equation (15), we have

$$\mathbf{p} = -[(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{B} + \kappa(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{C}]\mathbf{q}\,. \tag{18}$$

Substituting (18) into (14), (16) and (17), we have

$$L = \mathbf{q}^t[\mathbf{D} - \lambda\mathbf{I}]\mathbf{q} + \lambda\,, \tag{19}$$

$$\frac{1}{2}\frac{\partial L}{\partial \mathbf{q}} = [\mathbf{D} - \lambda\mathbf{I}]\mathbf{q} = 0\,, \tag{20}$$

$$\frac{1}{2}\frac{\partial L}{\partial \kappa} = \mathbf{q}^t[\kappa\mathbf{T} + \mathbf{S}/2]\mathbf{q} = 0\,, \tag{21}$$

where

$$\mathbf{D} \equiv \left[\kappa^2\mathbf{T} + \kappa\mathbf{S} + \mathbf{R}\right], \tag{22.1}$$

$$\mathbf{R} \equiv \mathbf{B}^t\mathbf{B} - \mathbf{B}^t\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{B}, \quad \mathbf{T} \equiv \mathbf{C}^t\mathbf{C} - \mathbf{C}^t\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{C}, \tag{22.2}$$

and

$$\mathbf{S} \equiv \mathbf{C}^t\mathbf{B} - \mathbf{C}^t\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{B} + \mathbf{B}^t\mathbf{C} - \mathbf{B}^t\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{C}. \tag{22.3}$$

Notice that λ is the eigenvalue of **D**, and that $L\,|_{\frac{\partial L}{\partial \mathbf{p}}=0, \frac{\partial L}{\partial \mathbf{q}}=0, \frac{\partial L}{\partial \kappa}=0} = \lambda$, i.e., the minimal residual error is $\lambda_{\min} = \min\{\text{eigenvalues of } \mathbf{D}\}$. Given a κ, **D** is uniquely determined and so is the minimal residual error, $\lambda_{\min}(\kappa)$. When there is no 2D observation error and the initial estimates of u_0, v_0 and δ_u are accurate, the minimal residual error, $\lambda_{\min}$, is equal to zero. Then from (20), we have

$$\mathbf{T}^{-1}[\mathbf{D}-0\mathbf{I}]\mathbf{q} = \left[\kappa^2\mathbf{I} + \kappa\mathbf{T}^{-1}\mathbf{S} + \mathbf{T}^{-1}\mathbf{R}\right]\mathbf{q} = 0\,. \tag{23}$$

It follows that

$$\kappa(\kappa\mathbf{q}) = \left[-\mathbf{T}^{-1}\mathbf{R} \quad -\mathbf{T}^{-1}\mathbf{S}\right]\begin{bmatrix}\mathbf{q}\\ \kappa\mathbf{q}\end{bmatrix}, \tag{24}$$

and

$$\kappa(\mathbf{q}) = [0 \quad \mathbf{I}]\begin{bmatrix}\mathbf{q}\\ \kappa\mathbf{q}\end{bmatrix}. \tag{25}$$

It is now obvious that κ is the real eigenvalue of **K** where $\mathbf{K} \equiv \begin{bmatrix}0 & \mathbf{I}\\ -\mathbf{T}^{-1}\mathbf{R} & -\mathbf{T}^{-1}\mathbf{S}\end{bmatrix}$,

since

$$\kappa\begin{bmatrix}\mathbf{q}\\ \kappa\mathbf{q}\end{bmatrix} = \mathbf{K}\begin{bmatrix}\mathbf{q}\\ \kappa\mathbf{q}\end{bmatrix}. \tag{26}$$

But in practice, no real κ can be found by solving the eigenvalues of **K**, i.e., we will usually obtain a complex (impractical) κ that makes the residual error equal to zero. In our experience, we can choose to use the real part of the eigenvalue with the smallest absolute imaginary part as an estimate of κ, denoted as $\hat{\kappa}$. Of course, we can use $\hat{\kappa}$ as the initial guess, and perform the one dimensional nonlinear search to find the optimal κ. However, since the $\hat{\kappa}$ is accurate enough, the nonlinear search shows little gain according to our experience. Therefore, we omit the nonlinear search in the calibration procedure. Once $\hat{\kappa}$ is determined, the vector **q** can be obtained by selecting the eigenvector corresponding to the smallest eigenvalue of the matrix $\left[\hat{\kappa}^2\mathbf{T} + \hat{\kappa}\mathbf{S} + \mathbf{R}\right]$.

Substituting **q** into (18), we then have **p**. Using the fact that the nine $r_i's$, in the definition of $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$, are components of a rotation matrix, we can decompose the vectors **p** and **q**, and obtain the parameters $\mathbf{R}_C^O, t_1, t_2, t_3, f, u_0, v_0$ and δ_u given δ_v, refer [5] and [6].

III.2 The new camera calibration technique using coplanar calibration points.

When only coplanar calibration points are available, the whole calibration procedure is similar to the one that uses noncoplanar calibration points. But the decomposition of the parameters should be done carefully since the estimated results are more sensitive to noise. Without loss of generality, we can choose the $x-y$ plane of the **OCS** to be the plane where the calibration plane lies on, and all z 's in equation (7) vanish, thus

$$(1-\kappa\varrho^2)(u-u_0)[x \;\; y \;\; 1]\begin{bmatrix} r_7 \\ r_8 \\ t_3 \end{bmatrix} = [x \;\; y \;\; 1]\begin{bmatrix} r_1 f/\delta_u \\ r_2 f/\delta_u \\ t_1 f/\delta_u \end{bmatrix} \tag{27.1}$$

$$(1-\kappa\varrho^2)(v-v_0)[x \;\; y \;\; 1]\begin{bmatrix} r_7 \\ r_8 \\ t_3 \end{bmatrix} = [x \;\; y \;\; 1]\begin{bmatrix} r_4 f/\delta_v \\ r_5 f/\delta_v \\ t_2 f/\delta_v \end{bmatrix} \tag{27.2}$$

Similar to section III.1, we define

$$\mathbf{P}_1^C \equiv \begin{bmatrix} r_1 f/\delta_u + r_7 u_0 \\ r_2 f/\delta_u + r_8 u_0 \\ t_1 f/\delta_u + t_3 u_0 \end{bmatrix}, \quad \mathbf{P}_2^C \equiv \begin{bmatrix} r_4 f/\delta_v + r_7 v_0 \\ r_5 f/\delta_v + r_8 v_0 \\ t_2 f/\delta_v + t_3 v_0 \end{bmatrix}, \quad \mathbf{P}_3^C \equiv \begin{bmatrix} r_7 \\ r_8 \\ t_3 \end{bmatrix}, \tag{28}$$

and we have

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

$$+\,\kappa\begin{bmatrix} \vdots & \vdots & \vdots \\ (u_j-u_0)\varrho_j^2 x_j & (u_j-u_0)\varrho_j^2 y_j & (u_j-u_0)\varrho_j^2 \\ (v_j-v_0)\varrho_j^2 x_j & (v_j-v_0)\varrho_j^2 y_j & (v_j-v_0)\varrho_j^2 \\ \vdots & \vdots & \vdots \end{bmatrix}\mathbf{P}_3^C = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{29}$$

To estimate the parameters of camera, the procedures similar to section III.1 are repeated, except that the definitions of matrix **A**, **B**, **C** should be replaced by

$$\mathbf{A}^C \equiv \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_j & y_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \quad \mathbf{B}^C \equiv \begin{bmatrix} \vdots & \vdots & \vdots \\ -u_jx_j & -u_jy_j & -u_j \\ -v_jx_j & -v_jy_j & -v_j \\ \vdots & \vdots & \vdots \end{bmatrix},$$

$$\mathbf{C}^C \equiv \begin{bmatrix} \vdots & \vdots & \vdots \\ (u_j-\hat{u}_0)\hat{\varrho}_j^2x_j & (u_j-\hat{u}_0)\hat{\varrho}_j^2y_j & (u_j-\hat{u}_0)\hat{\varrho}_j^2 \\ (v_j-\hat{v}_0)\hat{\varrho}_j^2x_j & (v_j-\hat{v}_0)\hat{\varrho}_j^2y_j & (v_j-\hat{v}_0)\hat{\varrho}_j^2 \\ \vdots & \vdots & \vdots \end{bmatrix},$$

and the decomposition procedure should be modified as follows. Suppose that the estimated composite parameters are $\mathbf{P}_1^C = t_3 \cdot [a_1 \;\; a_2 \;\; a_3]^t$, $\mathbf{P}_2{}^C = t_3 \cdot [a_4 \;\; a_5 \;\; a_6]^t$ and $\mathbf{P}_3{}^C = t_3 \cdot [a_7 \;\; a_8 \;\; 1]^t$, where $a_i, i = 1 \ldots 8$, are real numbers. Substituting these values into the definitions of $\mathbf{P}_1^C$, $\mathbf{P}_2^C$ and $\mathbf{P}_3^C$, i.e. equation (28), we have the following eleven equations of eleven unknowns: $r_1,\;\; r_2,\;\; r_4,\;\; r_5,\;\; r_7,\;\; r_8,\;\; t_1,\;\; t_2,\;\; t_3,\;\; \delta_u$ and f,

$$\begin{cases} r_1 = (a_1 - a_7u_0)\delta_u t_3/f \\ r_2 = (a_2 - a_8u_0)\delta_u t_3/f \\ t_1 = (a_3 - u_0)\delta_u t_3/f \\ r_4 = (a_4 - a_7v_0)\delta_v t_3/f \\ r_5 = (a_5 - a_8v_0)\delta_v t_3/f \\ t_2 = (a_6 - v_0)\delta_v t_3/f \\ r_7 = a_7t_3 \\ r_8 = a_8t_3 \\ r_1^2 + r_4^2 + r_7^2 = 1 \\ r_2^2 + r_5^2 + r_8^2 = 1 \\ r_1r_2 + r_4r_5 + r_7r_8 = 0 \end{cases} \qquad (30)$$

where the last three equations are the constraints of a rotation matrix. Substituting the first eight equations to the last three, we have

$$\begin{bmatrix} (a_1-a_7u_0)^2 & (a_4-a_7v_0)^2 & a_7^2 \\ (a_2-a_8u_0)^2 & (a_5-a_8v_0)^2 & a_8^2 \\ (a_1-a_7u_0)(a_2-a_8u_0) & (a_4-a_7v_0)(a_5-a_8v_0) & a_7a_8 \end{bmatrix} \begin{bmatrix} (\delta_u t_3/f)^2 \\ (t_3/f)^2 \\ t_3^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. \qquad (31)$$

By solving equation (31), t_3, δ_u and f can be found, since they are known to be positive numbers. Knowing the three values of t_3, δ_u and f, all the extrinsic parameters in equation

(30) can be easily obtained. Finally, the last column of the rotation matrix can be obtained by calculating the cross product of the first two columns, i.e.,

$$\begin{bmatrix} r_3 \\ r_6 \\ r_9 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_4 \\ r_7 \end{bmatrix} \times \begin{bmatrix} r_2 \\ r_5 \\ r_8 \end{bmatrix}. \tag{32}$$

IV. EXPERIMENTAL RESULTS

In this section, we will show some experimental results obtained by both computer simulations and real experiments. To evaluate the accuracy of the camera calibration for 3D vision application, it is necessary to define certain kinds of error measure. The measure used in this report is the *3D angular error*, i.e, the angle $\sphericalangle\, POP'$ shown in Fig. 3, where P is the 3D test point, O is the estimated lens center, and OP' is the 3D ray back projected from the observed 2D image of P using the estimated camera parameters. A 3D angular error of 0.005^o is roughly equivalent to "1 part in 10000", because $tan(0.005^o) \approx 1/10000$.

For convenience, let $\beta_{L\kappa}$, β_T, and β_N denote the estimate of β obtained respectively by our algorithm, Tsai's two-stage algorithm[8], and Weng's two-step nonlinear algorithm

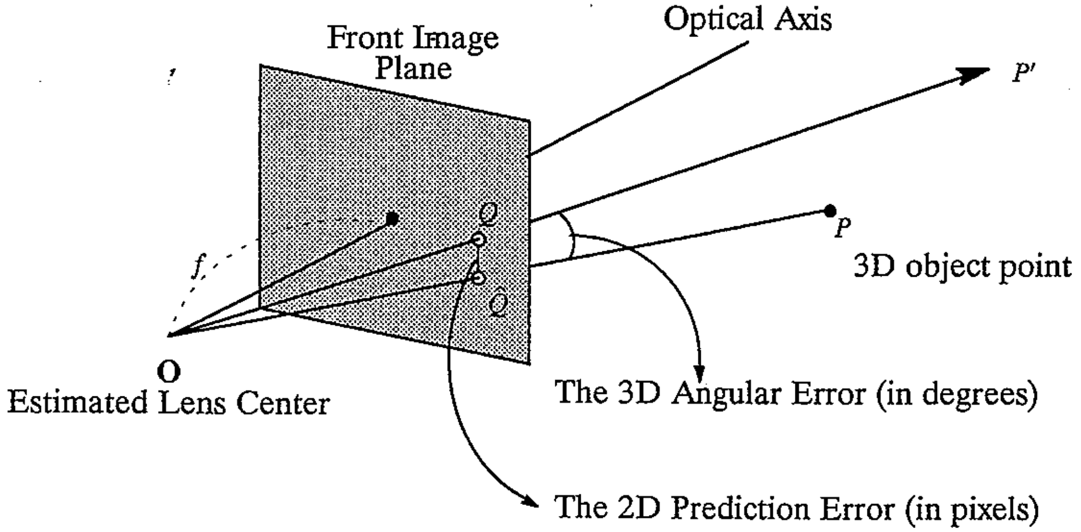


Fig. 3 The definition of 3D angular error.

(only radial lens distortion is considered)[9]. To compute $\beta_{L\kappa}$ and β_T, it is required that u_0 and v_0 are known *a priori*. In the experiment we can set the initial value for u_0 and v_0 to be zero, and the initial value for δ_u is $\delta_u' \cdot f_{camera}/f_{digitizer}$, for the reasons explained in section III. The image size is 480 × 512 pixels. In the simulation, the synthetic camera is assumed to have the true focal length $f = 25.85$ mm, and $\delta_u = 15.66$ µm, $\delta_v = 13$ µm (the initial value for δ_u is set to be $\delta_u' \cdot f_{camera}/f_{digitizer} = 11$ µm*14.31818 MHz/10 MHz, which introduces about 0.58% error for δ_u). We also assume the 3D positions of the calibration points are known exactly, and the only source of measurement error is the error in estimating the image coordinates of the calibration points, i.e., the 2D observation error. Each data shown in Figs. 4, 5 and 7 – 15 is an average of ten random trials. Let σ denote the standard deviation of the 2D observation error.

IV.1 Performance of different techniques using noncoplanar calibration points.

The first experiment is to determine a proper number of calibration points. In the computer simulation, it is assumed that a camera with small radial lens distortion (e.g. $\kappa = 0.0003$, which corresponds to roughly 2 to 3 pixels near the four corners of the image) is to be calibrated, and σ is assumed to be 0.1 pixel. Fig. 4 shows how the 3D angular error decreases as the number of calibration points, N_{calib}, increases. When the number of calibration points is greater than fifty, it shows little gain to increase N_{calib}. Therefore, we choose to use $N_{calib} = 60$ in the following experiments. Fig. 5 shows the results obtained from 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 for 25 mm. A typical image is shown in Fig. 6. Thus we have $21 \times 25 = 525$ pairs of 2D-3D coordinates of control points. We randomly choose N_{calib}

Noncoplanar Calibration points

Computer Simulation: $\sigma = 0.1$ *pixel*, $\kappa = +0.0003$ mm^{-2}

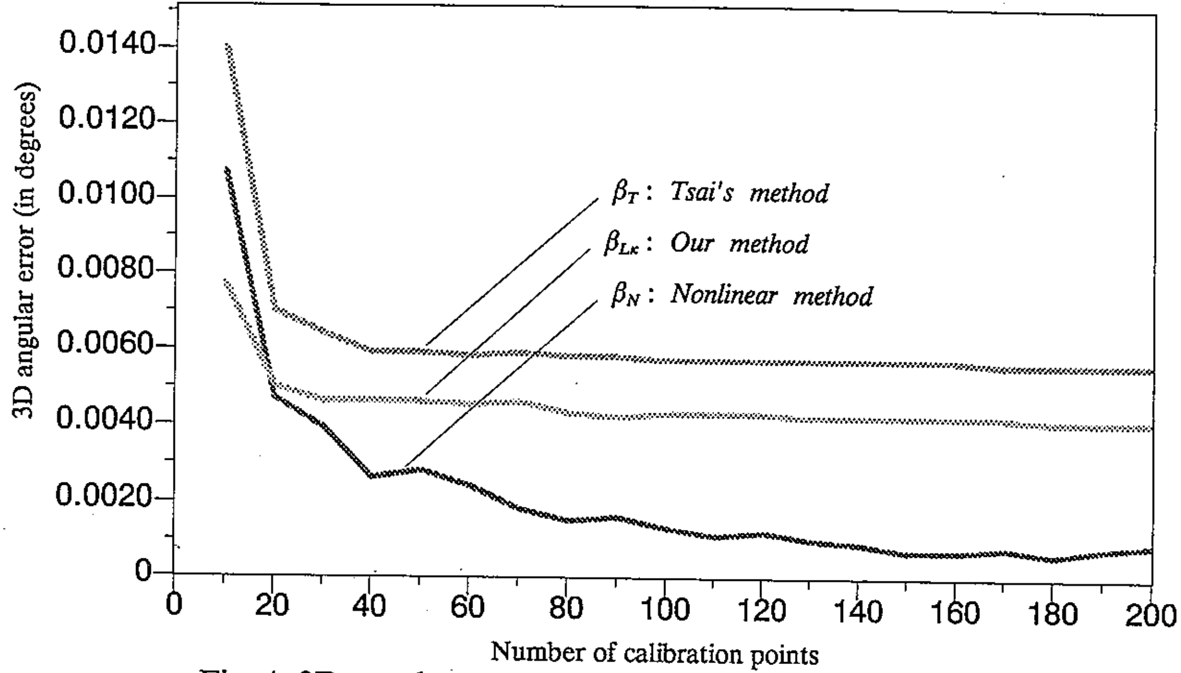


Fig. 4 3D angular error versus the number of calibration points

Noncoplanar Calibration points

Real Experiment

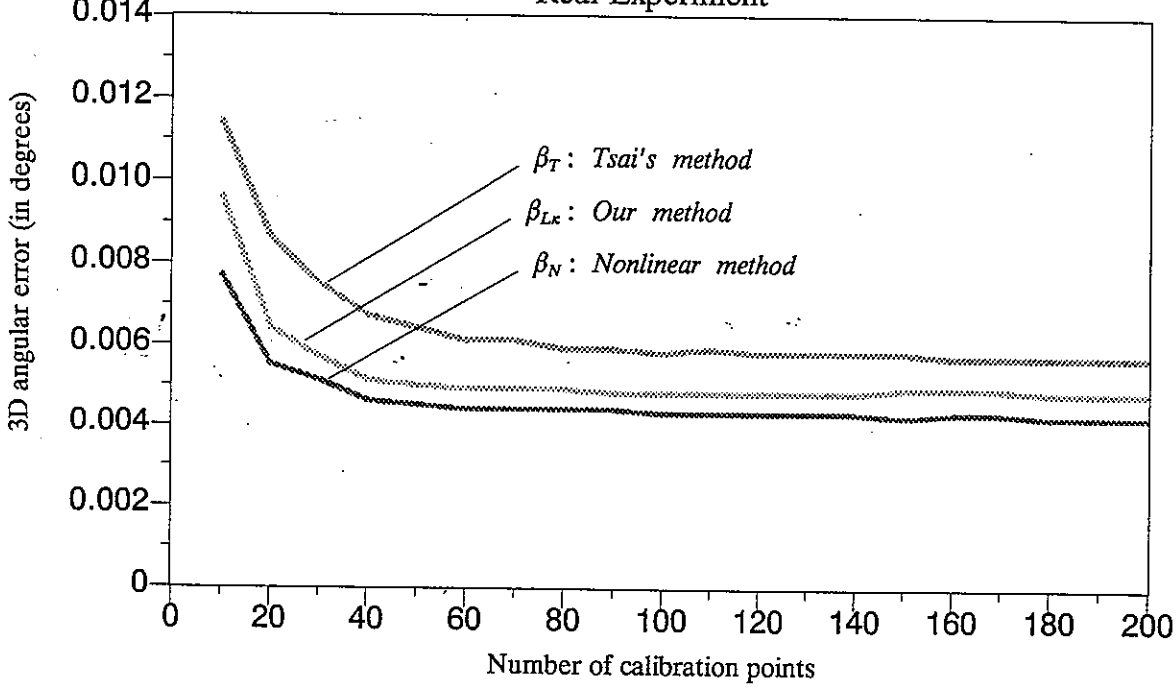


Fig. 5 3D angular error versus the number of calibration points

points from the 525 2D-3D pairs to calibrate the camera and use all remaining points to test the calibrated parameters. Since the 3D coordinates of the test points used *in the real*

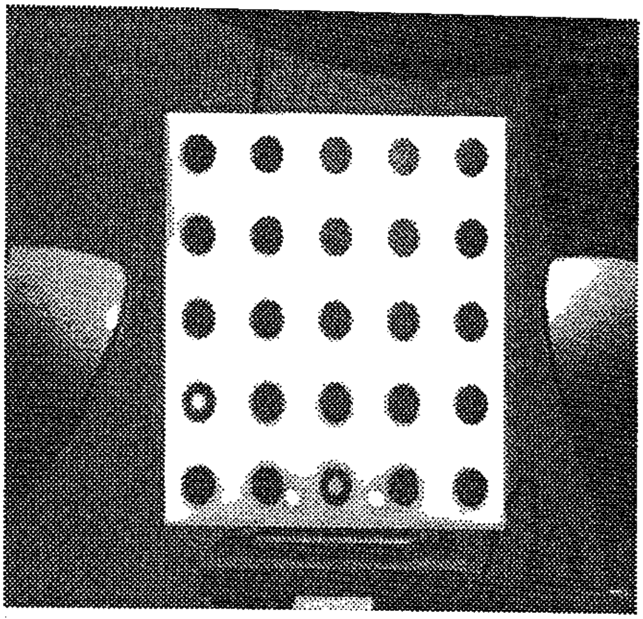


Fig. 6 A typical image of the calibration plate containing 25 calibration points used in the real experiment.

experiment also contain errors, the computed 3D angular errors are larger than those from the computer simulation.

Next simulation shows the deterioration of the three calibration techniques as the 2D observation errors increase (see Fig. 7). The 3D angular error of Tsai's technique is always greater than the other two. This is partially because, each 2D-3D pair of the calibration points contributes only one equation when using the radial alignment constraint[8], while the other two techniques use both the horizontal and vertical components of the perspective projection relation and each 2D-3D pair contributes two equations in the minimization. Another reason is that the Tsai's algorithm[8] does not estimate the image center. While the image center could be estimated via a separate procedure depicted in [9], we had not implemented that image center estimation algorithm. Instead, we showed that, given the same guess for image center, our method has better performance than that of Tsai's method. When given more accurate image center, both Tsai's and our techniques obtain better results, but our method

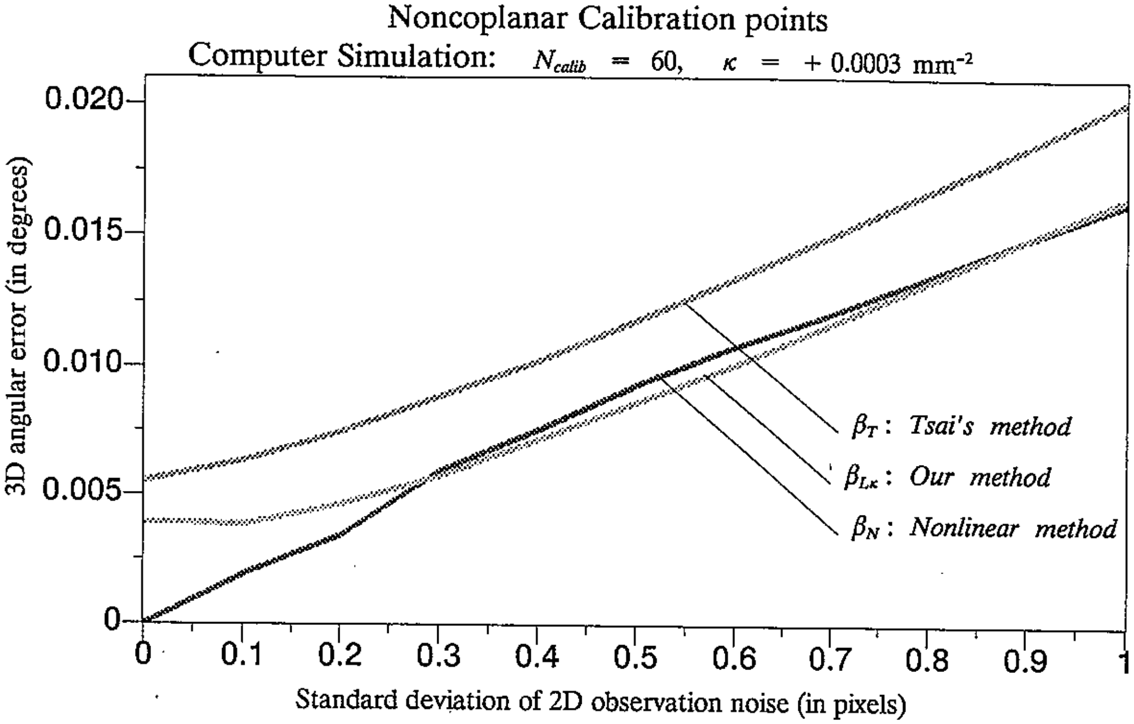


Fig. 7 3D angular error versus the 2D observation noise

still perform better than Tsai's does. Sometimes, our method even has better performance than Weng's method, because Weng's method can be trapped in a local minimum.

In the third experiment, we set the 2D observation error $\sigma = 0.1$ and the number of calibration points $N_{calib} = 60$, and observe how the 3D angular error varies versus the radial lens distortion coefficient, as shown in Fig. 8. Notice that both Tsai's (β_T) and our method ($\beta_{L\kappa}$) degrade when κ becomes larger. This is caused by being given a wrong image center. Also, Fig. 8 displays a phenomenon that we have predicted in section III, i.e., our technique will have better performance when the radial lens distortion is smaller. When the given image center is not accurate and $\kappa > 0$, our technique tends to over-estimate the κ. Thus for a positive and large κ (rare in practice though), our technique may give a worse estimate than those given by others if the initial estimate for the image center is not good enough. However, in the application of computer vision, we often choose to use a standard lens ($|\kappa| \approx 0$) or a wide angle lens ($\kappa < 0$), rather than a lens with $\kappa \gg 0$. Therefore, the proposed technique is

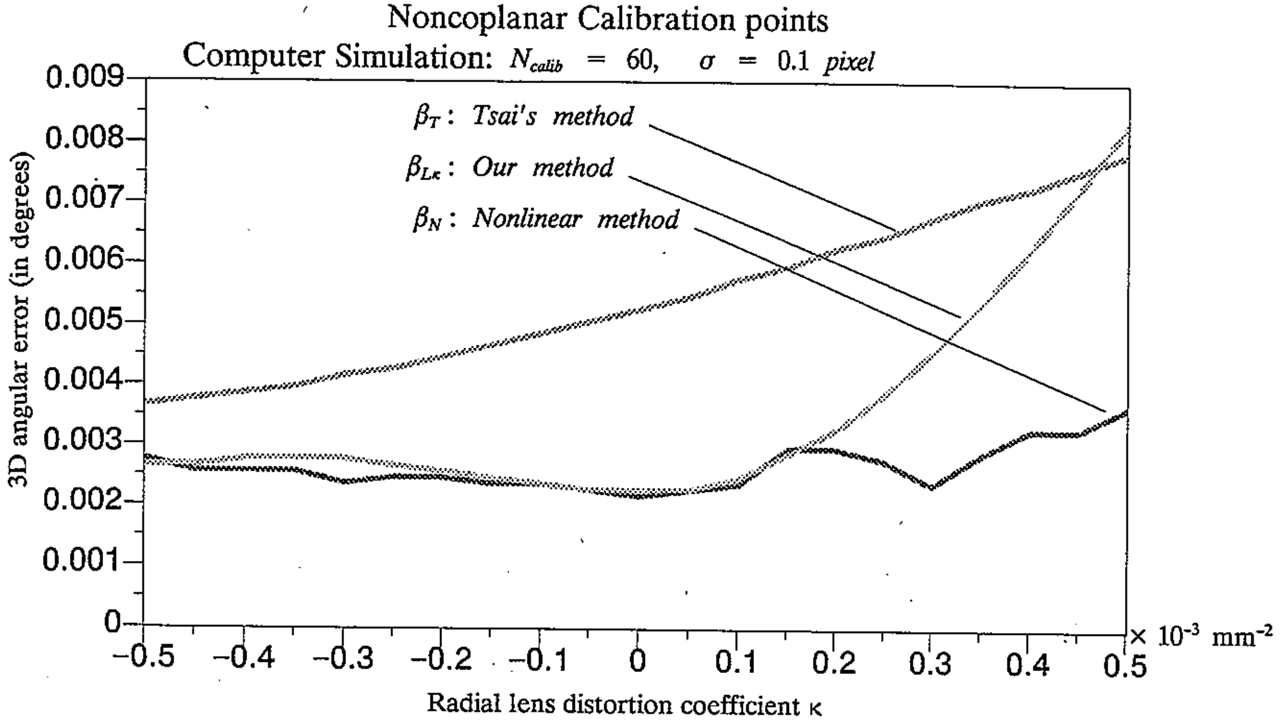


Fig. 8 3D angular error versus the radial lens distortion coefficient

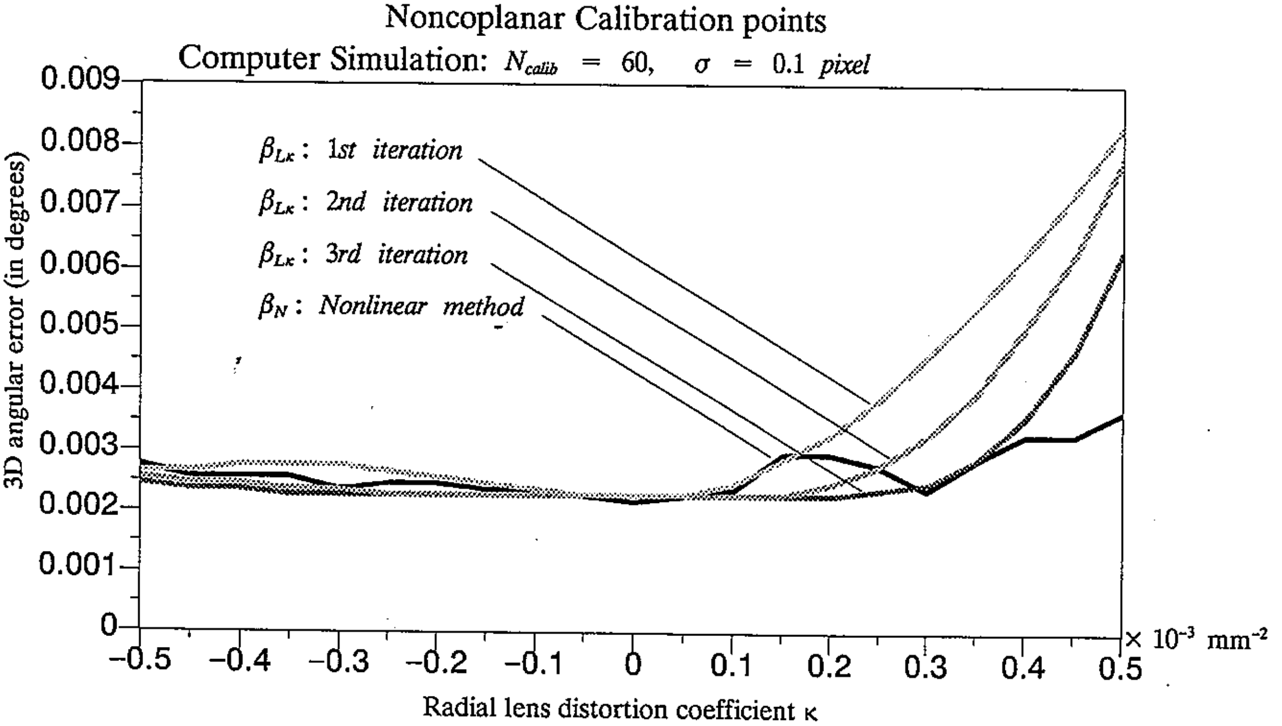


Fig. 9 3D angular error versus the radial lens distortion coefficient

suitable for most 3D computer vision applications requiring high accuracy. Using our method, the calibration procedure can be executed iteratively by substituting the originally given image center (u_0, v_0) and δ_u with the newly estimated ones. The first three iterations are shown in Fig. 9. Notice that the performance of the nonlinear method that typically needs 40–100 iterations is sometimes even worse than that obtained after the second (or the third) iteration of our method.

The forth simulation shows how Tsai's method and our method degrade as the error of the given image center becomes large. In this simulation, the true image center is $(R_{Center}cos(a),\ R_{Center}sin(a))$, and the *a priori* given image center is set to $(0, 0)$. Each data point shown in Figs. 10 and 11 is the average of forty random trials, ten for each angle of $a = \pm\ 45,\ \pm\ 135$. Notice that when the given image center is less accurate, both Tsai's (β_T) and our method ($\beta_{L\kappa}$) degrade more when κ becomes larger. If we use the true image center for the simulation, then the two curves corresponding to β_T and $\beta_{L\kappa}$ will be quite flat.

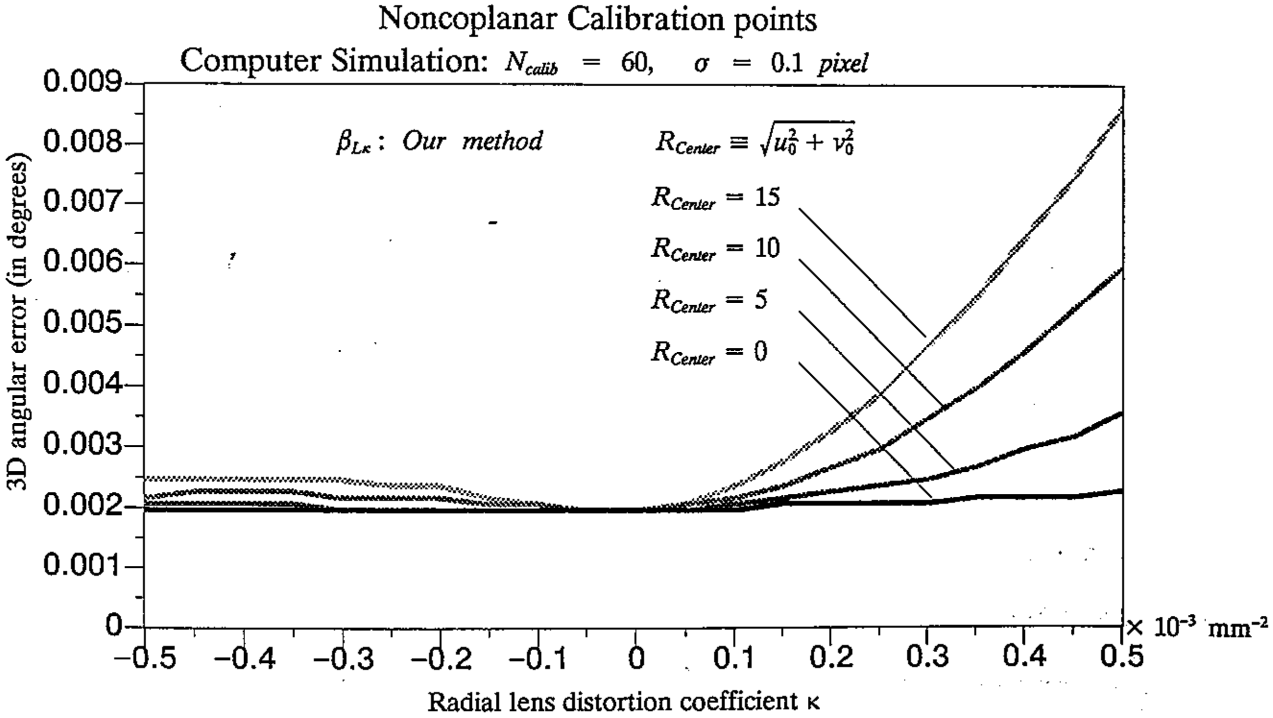


Fig. 10 3D angular error versus the radial lens distortion coefficient

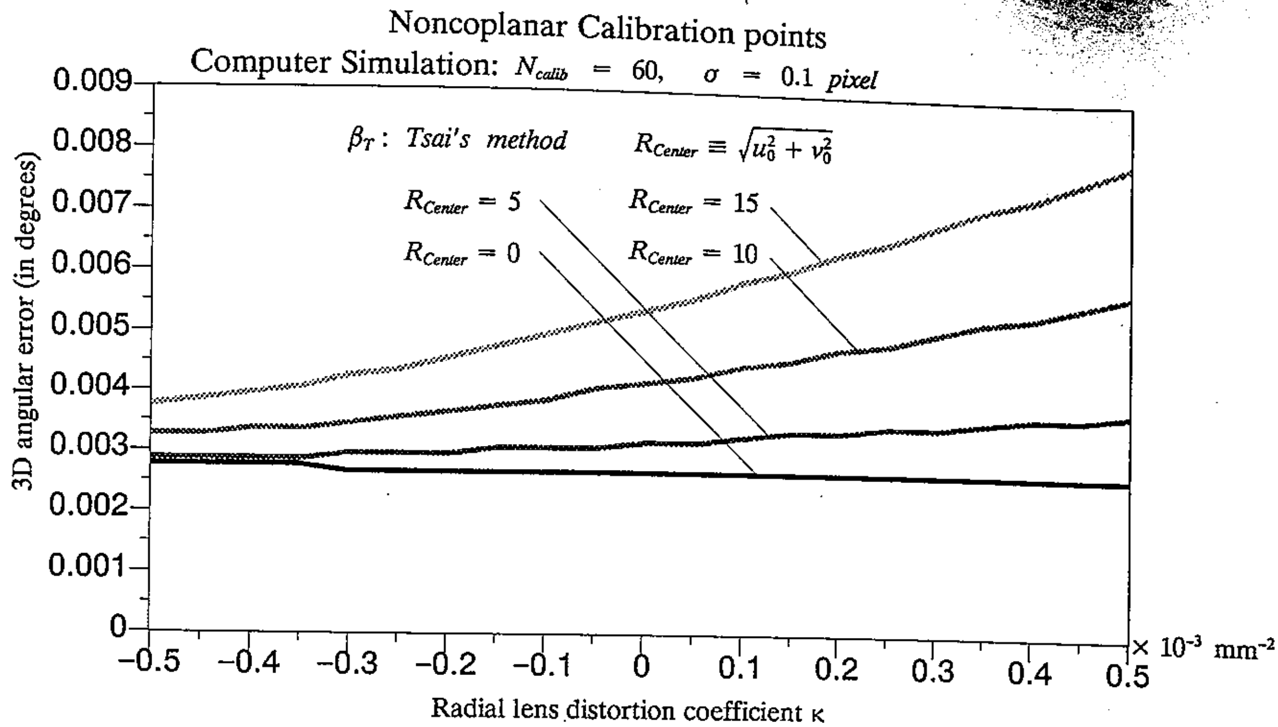


Fig. 11 3D angular error versus the radial lens distortion coefficient

IV.2 Performance of different techniques using coplanar calibration points.

In section III.2 we have proposed a method for calibrating a camera using coplanar calibration points. Since the calibration technique using coplanar calibration points (coplanar technique, for short) behaves quite differently comparing to that using noncoplanar one (noncoplanar technique, for short), we discuss the experiments about coplanar techniques in this section. We set the 2D observation error $\sigma = 0.1$ and the number of calibration points $N_{calib} = 60$ for the following experiments. It should be emphasized that both Tsai's and our coplanar techniques do not estimate the image center, but our method gives an estimate of δ_u while Tsai's method does not.

We first show that our coplanar technique is more sensitive to the given image center comparing to Tsai's coplanar technique. In this simulation, the true image center is $(R_{Center}cos(a), \ R_{Center}sin(a))$, while the image center given to the algorithm is $(0, 0)$. Each data point shown in Figs. 12 – 14 is the average of forty random trials, ten for each angle of

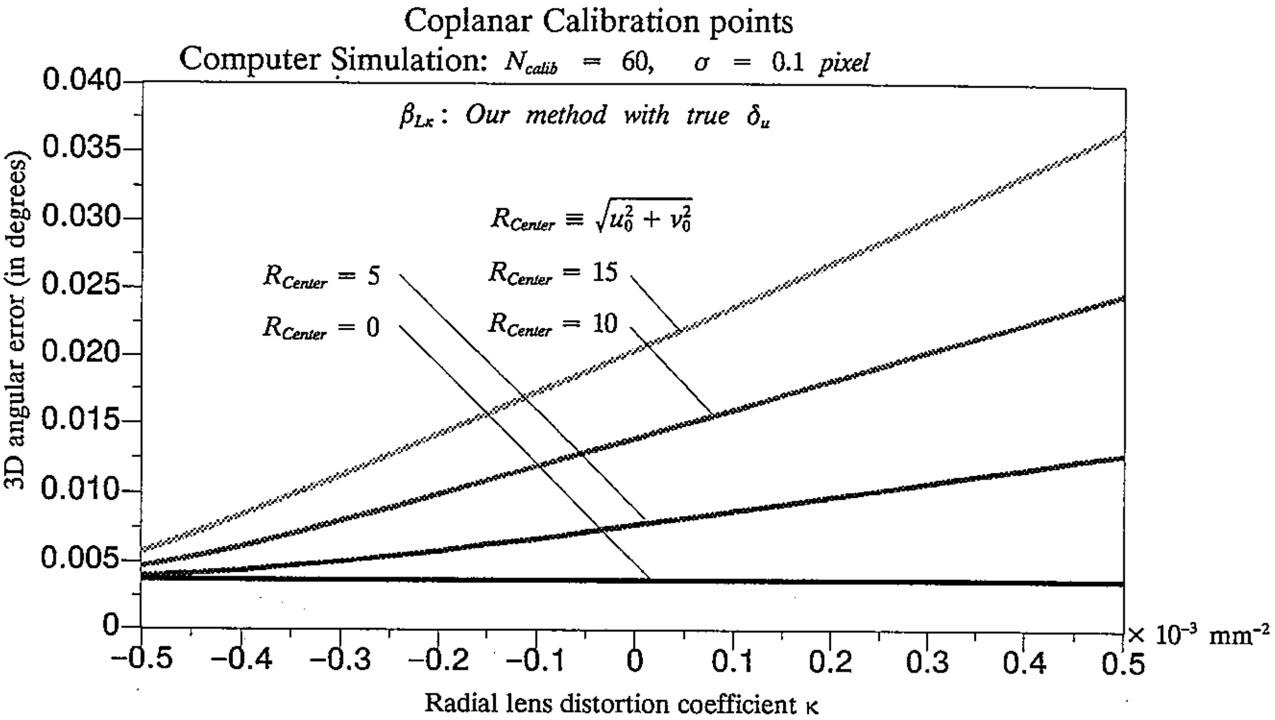


Fig. 12 3D angular error versus the radial lens distortion coefficient using our method with true δ_u

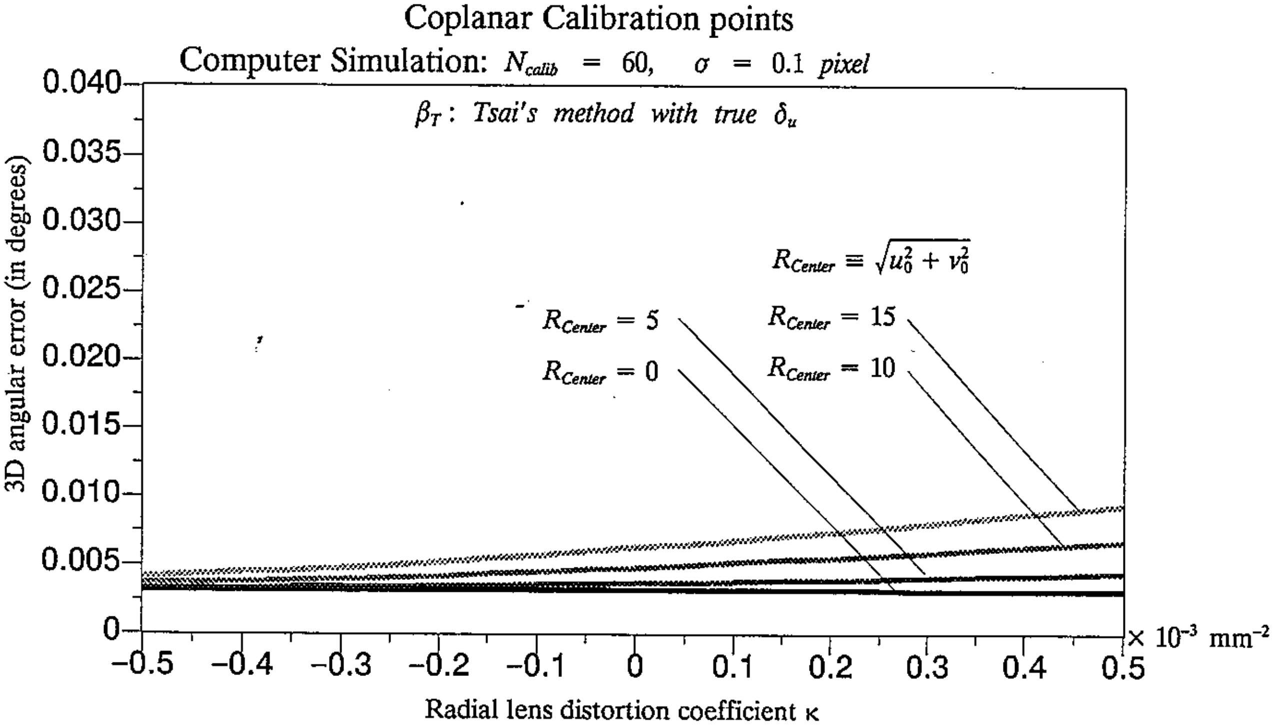


Fig. 13 3D angular error versus the radial lens distortion coefficient using Tsai's method with true δ_u

$\alpha = \pm 45, \pm 135$. Fig. 12 and Fig. 13 are the results obtained by using the true δ_u ; Fig. 14

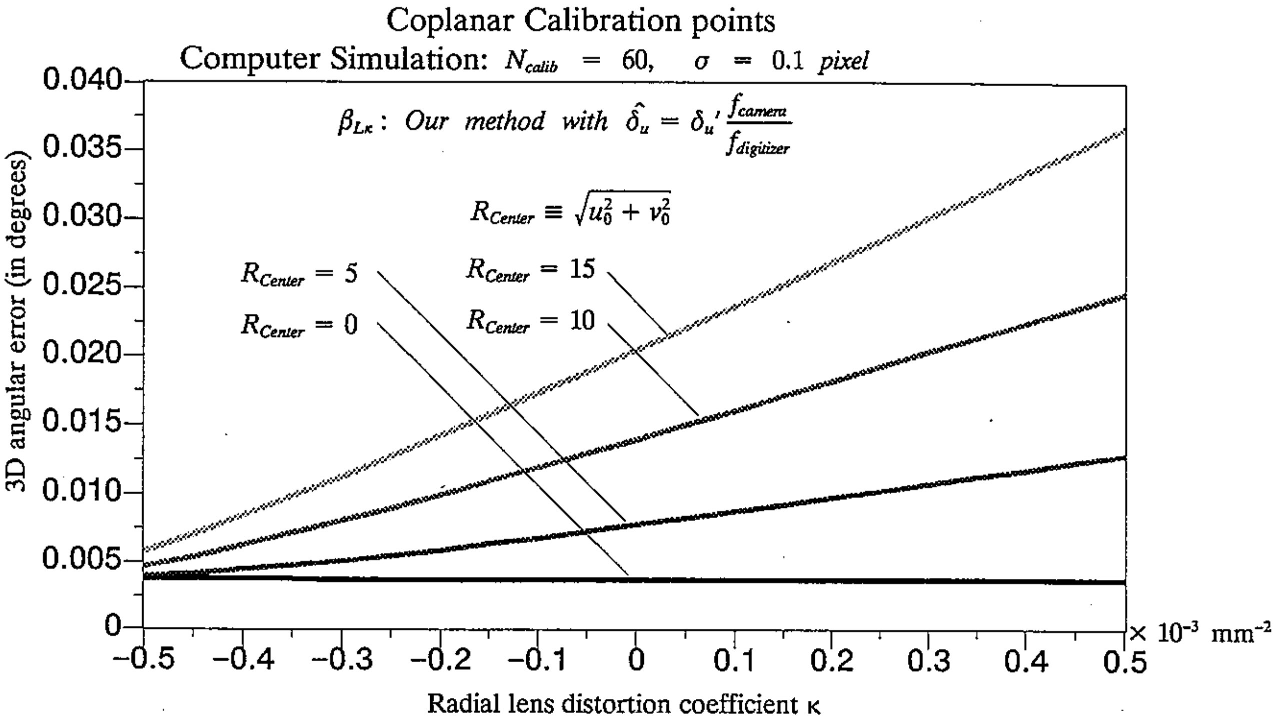


Fig. 14 3D angular error versus the radial lens distortion coefficient using our method with an estimate of δ_u having 0.58% of error

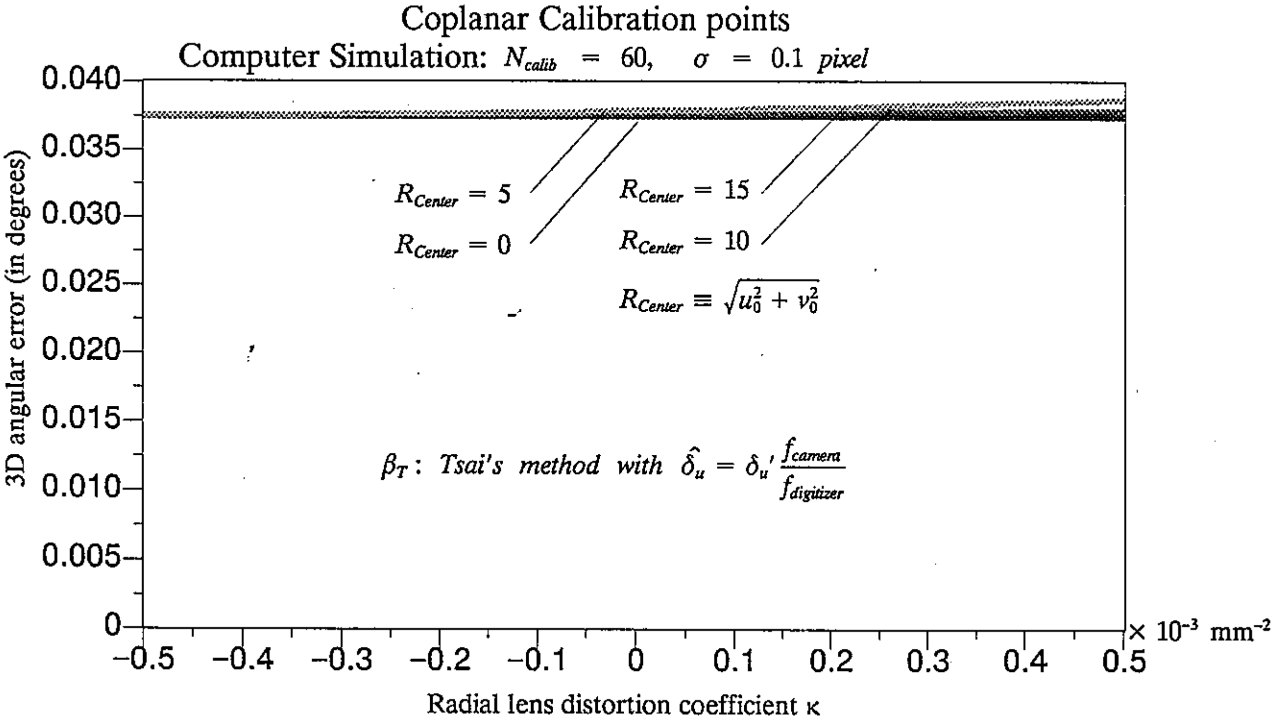


Fig. 15 3D angular error versus the radial lens distortion coefficient using Tsai's method with an estimate of δ_u having 0.58% of error

and Fig. 15 are obtained by using equation (6) as an estimate, which has the relative error

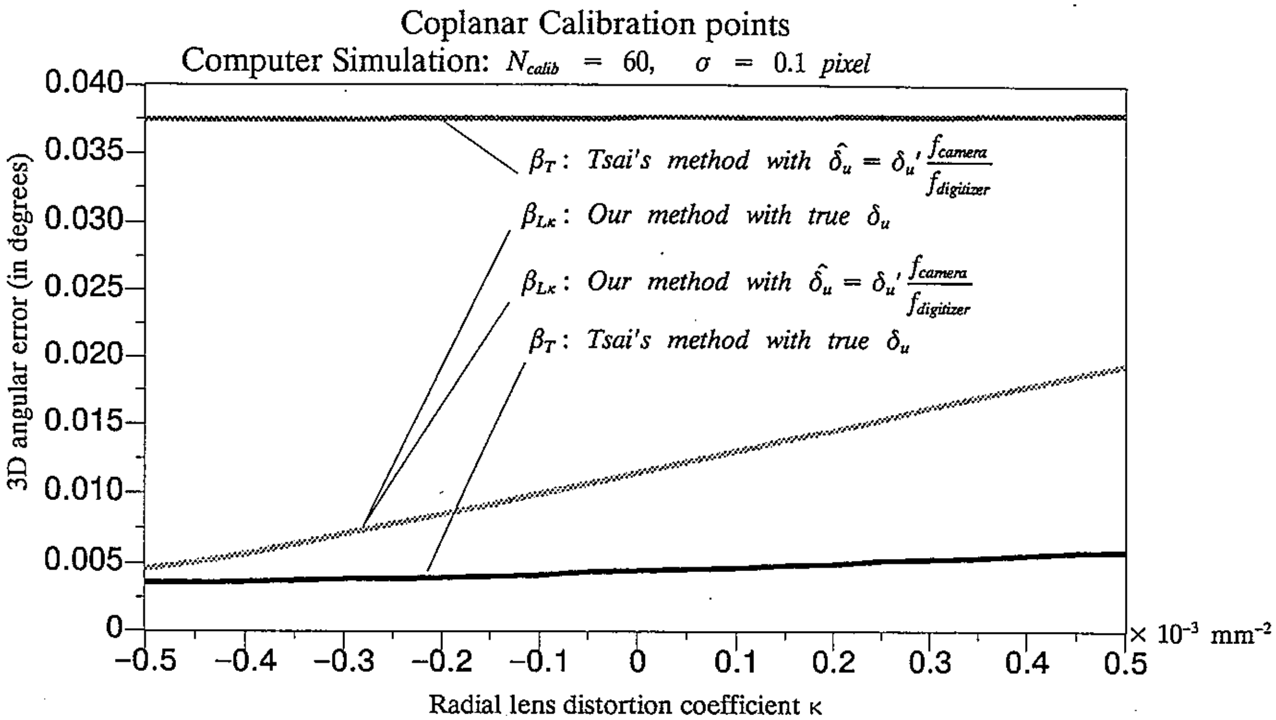


Fig. 16 3D angular error versus the radial lens distortion coefficient

about 0.58%, of δ_u. Notice that our coplanar technique is more robust when the given estimate of the horizontal pixel spacing δ_u is not exactly the true value, while Tsai's method is more robust to the deviation of the given estimate of the image center (u_0, v_0). The former is because we estimate δ_u, and the latter is because what we obtain from our linear calibration procedure are the composite parameters. The composite parameters are the redundant combination of the real parameters, which means an erroneous combination of these parameters can still make a good fit between experimental observations and model prediction[8].

Also, it is interesting to notice that the Tsai's coplanar technique has the performance which is almost equal to the Tsai's noncoplanar technique, if the given δ_u is accurate (see Fig. 13). However, the performance of Tsai's coplanar technique is sensitive to the accuracy of the given δ_u (see Fig. 15, a 0.58% error of δ_u will induce a large error on the calibration result).

Each curve in Fig. 16 is an average of the four curves in each of the four figures from Fig. 12 to Fig. 15. This figure shows that, unless we have an a very precise estimate of δ_u [10], it is better to use our calibration technique even when only coplanar calibration points are available.

V. CONCLUDING REMARKS

In this report, we have described a new camera calibration technique which is computationally fast and can achieve very high calibration accuracy for 3D computer vision applications. Our method is fast since it requires only linear computation and does not require any iterations. With a SPARCstation, the computation for the calibration can be done within a fraction of second. Moreover, our method can achieve very high accuracy for 3D estimation (i.e. small 3D angular error) because the effect of the lens distortion is considered and all the information contained in the calibration points are used. We have shown that our new noncoplanar calibration method can achieve the 3D angular error of 0.005^o, or the accuracy of "1 part in 10000" in 3D measurement. This means that we will make only 0.1mm of error in 3D position estimation when the object is one meter away from the camera. For practical reasonable lens distortion, our linear method is good enough for almost all the 3D applications, and further nonlinear iteration is not necessary. Also, it may not worth the efforts to increase the calibration accuracy using nonlinear minimization techniques unless we can increase the accuracy of 2D feature extraction accordingly.

REFERENCE

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

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

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

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

[5] Y.P. Hung, "Three Dimensional Surface Recontruction Using a Moving Camera A Model–Based Probabilistic Approach," Ph.D dissertation, Division of Engineering, Brown University, Providence, R.I. 1990; also, Technical Report LEMS–63, 1989

[6] T.M. Strat, "Recovering the Camera Parameters from a Transformation Matrix," DARPA Image Understanding Workshop, 1984, pp. 264–271.

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

[8] 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.

[9] 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.

[10] R.K. Lenz, R.Y. Tsai, "Techniques for Calibration of theScale Factor and Image Center for High Accuracy 3–DMachine Vision Metrology," IEEE Trans. Pattern Anal. Machine Intell., Vol. 10, No. 5, Sep. 1988, pp. 713–720.

[11] W.I. Grosky, L.A. Tamburino, "A Unified Approach to the Linear Camera Calibration Problem," IEEE Trans. Pattern Anal. Machine Intell., Vol. 12, NO. 7, July 1990, pp. 663–671.

[12] 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.

[13] K. Kanatani, Y. Onodera, "Noise Robust Camera Calibration Using Vanishing Points," IEICE Transactions on Information and Systems, Vol. E74, No. 10, Oct. 1991.