Machine Vision and Applications (2006) 17(1): 51–67
DOI 10.1007/s00138-006-0014-6

ORIGINAL PAPER

Wei Sun · Jeremy R. Cooperstock

An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques

Received: 23 November 2004 / Accepted: 3 January 2006 / Published online: 25 February 2006

Abstract This paper presents an empirical study investigating the effects of training data quantity, measurement error, pixel coordinate noise, and the choice of camera model, on camera calibration accuracy, based on three publicly available techniques, developed by Tsai, Heikkilä and Zhang. Results are first provided for a simulated camera system and then verified through carefully controlled experiments using real-world measurements. Our aims are to provide suggestions to researchers who require an immediate solution for camera calibration and a warning of the practical difficulties one may encounter in a real environment. In addition, we offer some insight into the factors to keep in mind when selecting a calibration technique. Finally, we hope that this paper can serve as an introduction to the field for those newly embarking upon calibration-related research.

Keywords Camera calibration · Camera parameters · Coordinate transformations · Distortion models · Accuracy evaluation

1 Introduction

The decreasing cost of computers and cameras has brought rapid growth in stereo-based applications. In consequence, an ever-increasing population of researchers are coming to depend on camera calibration for their projects.

Although camera calibration is a mathematically well-defined problem and numerous methods have been developed to provide solutions, these methods make different simplifying assumptions as to how they should be used in practice and which variables need to be measured. It is also not clear how certain factors such as training data quantity, measurement error, and the choice of camera model influence calibration accuracy. The effect of noise has been studied by Lavest et al. [1] and Zhang [2]; however, their experiments involved a small volume of 3D space and did not use separate test data to verify the scalability of calibrated results. Moreover, a major practical concern is the degree of effort involved in providing a given camera calibration algorithm with the training data it needs to achieve some required accuracy.

Using three publicly available techniques, through extensive experimentation with separate training and test data, we conducted an empirical study of the impact of noise, either in world or pixel coordinates, and training data quantity, on calibration accuracy. We also included a detailed comparison of various models to determine the relative importance of different distortion components, whether the addition of higher-order radial terms or decentering distortions actually improves calibration accuracy, and if so, to what degree. Our contributions are first, to provide a quick answer to researchers who need an immediate camera calibration solution; and second, to give some insight into practical difficulties one may encounter in a real environment, and which factors to keep in mind when looking for a suitable calibration technique. Finally, we hope to offer an introduction to the field for those newly embarking upon calibration related research.

The calibration methods chosen for experimentation were developed by Tsai [3], Heikkilä [4] and Zhang [5]. An important reason for this choice is that the source code for the three methods is publicly available, well developed, well tested, and therefore provides a fair comparison. In fact, only open source algorithms were serious candidates for a study of this purpose, because they are open to study and readily integrated into a larger system.

Tsai's method represents a conventional approach that is based on the radial alignment constraint (RAC) and requires accurate 3D coordinate measurement with respect to a fixed reference. Among the conventional calibration methods surveyed by Salvi et al. [6], including those developed by Hall et al. [7], Faugeras–Toscani [8], and Weng et al. [9], Tsai's was reported to exhibit the best performance. His method

W. Sun (✉) · J. R. Cooperstock
Department of Electrical and Computer Engineering, Centre for Intelligent Machines, McGill University, 3480 University Street, Montreal, Quebec H3A 2A7, Canada
E-mail: {wsun, jer}@cim.mcgill.ca
Fax: +1-514-398-7348

has been widely used in multi-camera applications [10, 11]. Heikkilä's method, also world-reference based, although not included in that survey, employs the more general direct linear transformation (DLT) technique by making use of the prior knowledge of intrinsic parameters. It also involves a more complete camera model for lens distortions. Zhang's method is a different special case of Heikkilä's formulation. It combines the benefits of world-reference based and auto-calibration approaches, which enables the linear estimation of all supposedly constant intrinsic parameters. This method is flexible in that either the camera or the planar pattern can be moved freely and the calibration procedure is easily repeatable without redoing any measurements. These three methods of course form part of a large space of potential algorithms and can each be generalized in a variety of ways. We have applied the researchers' names to particular forms of their algorithms as embodied in published code.

2 Calibration methods

This section first provides a brief review of the existing calibration literature, which focuses mainly on camera modeling and calibration algorithm development, and then explains the mathematical details of the methods we investigate, followed by our evaluation criteria.

2.1 A brief literature review

Camera calibration has received increased attention in the computer vision community during the past 2 decades [6, 12]. According to the nature of training data, existing calibration methods can be classified as coplanar or non-coplanar. The coplanar approaches perform calibration on data points limited to a planar surface of a single depth. These methods are either computationally complex or fail to provide solutions for certain camera parameters, such as the image center, the scale factor, or lens distortion coefficients [13]. In contrast, the non-coplanar approaches, to which our study is confined, use training points scattered in 3D space to cover multiple depths, and do not exhibit such problems.

Non-coplanar approaches fall into a number of categories. World-reference based calibration is a conventional approach requiring the 3D world coordinates and corresponding 2D image coordinates of a set of feature points [3, 4, 6, 8, 9]. The disadvantage of this approach is that either a well-engineered calibration object is required, or the environment has to be carefully set up to achieve accurate 3D measurements. Geometric-invariant-based methods use parallel lines and vanishing points as calibration features. Although world coordinate measurement is not required, special equipment may be necessary for measuring certain variables, for example, the ratio of focal lengths [14]. Implicit calibration methods [15] have no explicit camera model and may achieve high accuracy, but are computationally expensive because of the large number of unknown variables involved and they do not reveal the physical parameters of a camera. Auto-calibration or self-calibration approaches determine camera parameters directly from multiple uncalibrated views of a scene by matching corresponding features between views, despite unknown camera motion and even changes in some of the intrinsic parameters [16, 17]. Unfortunately, due to the difficulty of initialization [12], auto-calibration results tend to be unstable [18]. Planar auto-calibration [19] addresses this initialization difficulty by using multiple views of a planar scene, taking advantage of the fact that planes are simple to process and allow reliable and precise feature or intensity-based matching. Sturm–Maybank [20] and Zhang [5] both extend this idea by taking into account the relative geometric information between planar feature points, with Zhang's method estimating lens distortion coefficients, a factor not included in the former work. According to Sturm and Maybank's singularity analysis [20], degenerate situations can be easily avoided. Another extension of the multiplanar approach suggests the use of angles and length ratios on a plane but provides no experimental results [21].

It is worth noting that these multiplanar calibration methods differ from the coplanar methods reviewed by Chatterjee and Roychowdhury [13]. The methods described here rely on a planar calibration pattern positioned at various orientations. They exploit the coplanarity constraint on the points in each sample set to reduce or eliminate the need for 3D measurement, but still sample a 3D region. Similarly, a one-dimensional object can be positioned at various spots and in various orientations in a 3D space to provide non-coplanar points for calibration [22].

The choice of camera model [23], varying mainly in its characterization of lens distortion, may also influence calibration results. Tsai used the second order radial distortion model [3] while Zhang employed both the second- and fourth- order terms [5]. Heikkilä included two further decentering distortion components [4], while Lavest et al. even added the sixth-order radial term [1]. Weng et al. introduced a thin prism distortion whose coefficients could be merged with those of the decentering distortion in actual calibration [9]. Most camera models assume zero skewness, i.e., the angle between x and y image axes is 90° [3, 4, 9], but Lavest et al. [1] and Zhang [5] estimate skewness as a variable.

2.2 Calibration methods for experimentation

After reviewing these calibration methods, we chose Tsai [3], Heikkilä [4] and Zhang's [5] methods for experimentation. The mathematical details are now provided.

2.2.1 Camera model

Tsai, Heikkilä and Zhang's methods all use the pinhole projective model to map 3D scenes to the 2D camera image plane. Despite different formulations for lens distortion, the mapping between world and image points proceeds

Table 1 Coordinate transformations in Tsai, Heikkilä and Zhang's camera models

Tsai	Heikkilä	Zhang
$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{t}$	$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{t}$	$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{t}$
$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \frac{f}{Z_c} \begin{bmatrix} X_c \\ Y_c \end{bmatrix}$	$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \frac{f}{Z_c} \begin{bmatrix} X_c \\ Y_c \end{bmatrix}$	$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} X_c \\ Y_c \end{bmatrix}$
$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \left(1 + k_1^{(T)} r^2\right) \begin{bmatrix} x_d \\ y_d \end{bmatrix}$	$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \left(1 + k_1^{(H)} r^2 + k_2^{(H)} r^4\right) \begin{bmatrix} x_d \\ y_d \end{bmatrix}$	$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \left(1 + k_1^{(Z)} r^2 + k_2^{(Z)} r^4\right) \begin{bmatrix} x_u \\ y_u \end{bmatrix}$
where $r = \sqrt{{x_d}^2 + {y_d}^2}$	$+ \begin{bmatrix} 2p_1^{(H)} x_d y_d + p_2^{(H)} (r^2 + 2{x_d}^2) \\ p_1^{(H)} (r^2 + 2{y_d}^2) + 2p_2^{(H)} x_d y_d \end{bmatrix}$ where $r = \sqrt{{x_d}^2 + {y_d}^2}$	where $r = \sqrt{{x_u}^2 + {y_u}^2}$
$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} s_x/d_x & 0 & c_x \\ 0 & 1/d_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$	$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} s_x D_x & 0 & c_x \\ 0 & D_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$	$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$

through the same four transformations, from world coordinates (X_w, Y_w, Z_w), via camera coordinates (X_c, Y_c, Z_c), undistorted image coordinates (x_u, y_u), and distorted image coordinates (x_d, y_d), to pixel coordinates (x_p, y_p), as shown in Table 1.

The first transformation between world coordinates, (X_w, Y_w, Z_w), and camera coordinates, (X_c, Y_c, Z_c), is determined by camera position and orientation, expressed by the *extrinsic parameters*, $\mathbf{R}$ and $\mathbf{t}$, where $\mathbf{t} = [t_x\ t_y\ t_z]^T$ is a translation vector, and

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$$

is a 3×3 rotation matrix which can also be expressed in terms of the roll-pitch-yaw angles, $\mathbf{r}_\theta = [\theta_x\ \theta_y\ \theta_z]^T$, as

$$\mathbf{R} = \begin{bmatrix} \cos\theta_y \cos\theta_z & \sin\theta_x \sin\theta_y \cos\theta_z - \cos\theta_x \sin\theta_z & \cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z \\ \cos\theta_y \sin\theta_z & \sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z & \cos\theta_x \sin\theta_y \sin\theta_z - \sin\theta_x \cos\theta_z \\ -\sin\theta_y & \sin\theta_x \cos\theta_y & \cos\theta_x \cos\theta_y \end{bmatrix}.$$

The other three transformations from camera coordinates, (X_c, Y_c, Z_c), to pixel coordinates, (x_p, y_p), are controlled by internal camera configuration, expressed by the *intrinsic parameters*. In Tsai and Heikkilä's models, these are: the camera's principal point in pixels or the image center, (c_x, c_y); the effective focal length, f; the image scale factor, s_x; the (supposedly known) center-to-center distances between adjacent pixels in x and y directions, d_x and d_y, or their inverses, D_x and D_y; the coefficients of Tsai's radial distortion, $k_1^{(T)}$; and Heikkilä's radial and decentering distortions, $k_1^{(H)}$, $k_2^{(H)}$, $p_1^{(H)}$ and $p_2^{(H)}$. In Zhang's model, the intrinsic parameters are: the image center, (c_x, c_y); the pixel focal lengths along the x and y axes, α and β; the radial distortion coefficients, $k_1^{(Z)}$ and $k_2^{(Z)}$; and the skew parameter, γ, describing the relative skewness of the x and y image axes of a camera. The focal lengths in the three models are related to each other as follows:

$$\alpha = f s_x / d_x = f s_x D_x, \quad \beta = f / d_y = f D_y. \tag{1}$$

Note that precise values are not needed for d_x, d_y and D_x, D_y in Tsai and Heikkilä's models as any error will be compensated for by the focal length, f, and the image scale factor, s_x, after calibration.

2.2.2 Calibration algorithms

All the three calibration algorithms estimate an initial closed-form solution by solving an over-determined system of linear equations. The initial estimates then proceed through a non-linear optimization process, such as the standard Levenberg–Marquardt algorithm as implemented in Minpack [24], to minimize residual error. The objective function used in the optimization step is usually the error of distorted pixel coordinates, which will be described in detail in Sect. 2.3, Eq. (10). As the three algorithms differ mainly in their estimation of the closed-form solution, this section offers further detail as to how the initial solution is obtained in each algorithm.

In Tsai's algorithm, given the 3D world and 2D distorted image coordinates of $n \gg 7$ feature points and assuming

the camera's image center, $(c_x,\ c_y)$, to be the center pixel of a captured image, a linear equation is formed based on the radial alignment constraint (RAC):

$$\begin{bmatrix} y_{\mathrm{d}}X_{\mathrm{w}} & y_{\mathrm{d}}Y_{\mathrm{w}} & y_{\mathrm{d}}Z_{\mathrm{w}} & y_{\mathrm{d}} & -x_{\mathrm{d}}X_{\mathrm{w}} & -x_{\mathrm{d}}Y_{\mathrm{w}} & -x_{\mathrm{d}}Z_{\mathrm{w}} \end{bmatrix} \times \begin{bmatrix} t_y^{-1}s_x r_1 \\ t_y^{-1}s_x r_2 \\ t_y^{-1}s_x r_3 \\ t_y^{-1}s_x t_x \\ t_y^{-1} r_4 \\ t_y^{-1} r_5 \\ t_y^{-1} r_6 \end{bmatrix} = x_{\mathrm{d}}. \tag{2}$$

With n such equations, an over-determined linear system can be established to solve for the unknowns, $s_x, t_x, t_y, r_1, \ldots, r_6$. The third row of the rotation matrix, $r_7, \ldots, r_9$, can be computed as the cross product of the first two rows. With the same n feature points, the focal length, f, and the last element of the translation vector, t_z, can be estimated from another over-determined system of n equations of the following form:

$$\begin{aligned} &\begin{bmatrix} r_4X_{\mathrm{w}} + r_5Y_{\mathrm{w}} + r_6Z_{\mathrm{w}} + t_y & -d_y(y_{\mathrm{p}} - c_y) \end{bmatrix} \begin{bmatrix} f \\ t_z \end{bmatrix} \\ &\quad = d_y(y_{\mathrm{p}} - c_y)(r_7X_{\mathrm{w}} + r_8Y_{\mathrm{w}} + r_9Z_{\mathrm{w}}), \end{aligned} \tag{3}$$

where d_x and d_y are the distances between adjacent pixels in the x and y directions. The estimates of f and t_z and an assumption of zero for the radial distortion coefficient, $k_1^{(\mathrm{T})}$, then serve as an initial estimate for a Levenberg–Marquardt algorithm. Finally, all parameters derived from linear estimation and the image center, $(c_x,\ c_y)$, are optimized iteratively by the Levenberg–Marquardt algorithm to refine the solution.

Heikkilä's algorithm is based on the direct linear transformation (DLT) method [17], which is a general form of Tsai's algorithm. Ignoring nonlinear distortions, a linear projective transformation, $\mathbf{P}$, maps 3D world points, $(X_{\mathrm{w}}, Y_{\mathrm{w}}, Z_{\mathrm{w}})$, to their corresponding pixel points, $(x_{\mathrm{p}}, y_{\mathrm{p}})$, up to a scale, μ:

$$\mu[x_{\mathrm{p}}\ \ y_{\mathrm{p}}\ \ 1]^{\mathrm{T}} = \mathbf{P}[X_{\mathrm{w}}\ \ Y_{\mathrm{w}}\ \ Z_{\mathrm{w}}\ \ 1]^{\mathrm{T}} \tag{4}$$

where $\mathbf{P} = \mathbf{K}[\mathbf{R}\ \ \mathbf{t}]$, $\mathbf{R}$ and $\mathbf{t}$ are extrinsic parameters, and

$$\mathbf{K} = \begin{bmatrix} f s_x D_x & 0 & c_x \\ 0 & f D_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

determines the coordinate transformation between distorted image coordinates, $(x_{\mathrm{d}}, y_{\mathrm{d}})$, and pixel coordinates, $(x_{\mathrm{p}}, y_{\mathrm{p}})$, as shown in Table 1. Given $n \gg 5$ feature points, $\mathbf{P}$ is obtained from an over-determined system of n pairs of the following equation by eliminating μ:

$$\begin{bmatrix} X_{\mathrm{w}} & Y_{\mathrm{w}} & Z_{\mathrm{w}} & 1 & 0 & 0 & 0 & 0 & -x_{\mathrm{p}}X_{\mathrm{w}} & -x_{\mathrm{p}}Y_{\mathrm{w}} & -x_{\mathrm{p}}Z_{\mathrm{w}} & -x_{\mathrm{p}} \\ 0 & 0 & 0 & 0 & X_{\mathrm{w}} & Y_{\mathrm{w}} & Z_{\mathrm{w}} & 1 & -y_{\mathrm{p}}X_{\mathrm{w}} & -y_{\mathrm{p}}Y_{\mathrm{w}} & -y_{\mathrm{p}}Z_{\mathrm{w}} & -y_{\mathrm{p}} \end{bmatrix} \times \begin{bmatrix} \bar{\mathbf{p}}_{\mathbf{1}} \\ \bar{\mathbf{p}}_{\mathbf{2}} \\ \bar{\mathbf{p}}_{\mathbf{3}} \end{bmatrix} = 0, \tag{5}$$

where $\bar{\mathbf{p}}_{\mathbf{i}}$ $(i = 1, 2, 3)$ is the ith row vector of $\mathbf{P}$. Setting f to an initial estimate, s_x to 1, and $(c_x,\ c_y)$ to the center pixel of an image, the extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$ are computed by

$$\mathbf{R} = \mathbf{K}^{-1}[\mathbf{p_1}\ \ \mathbf{p_2}\ \ \mathbf{p_3}], \quad \mathbf{t} = \mathbf{K}^{-1}\mathbf{p_4}, \tag{6}$$

where $\mathbf{p_i}$ $(i = 1, 2, 3, 4)$ is the ith column vector of $\mathbf{P}$. The Levenberg–Marquardt algorithm is then applied to find exact values of the intrinsic parameters including the distortion coefficients, $k_1^{(\mathrm{H})}$, $k_2^{(\mathrm{H})}$, $p_1^{(\mathrm{H})}$, $p_2^{(\mathrm{H})}$.

In Zhang's calibration process, a planar calibration pattern is presented to a camera in various orientations and is always assumed to be on $Z_{\mathrm{w}} = 0$ of a changing world coordinate system. The calibration algorithm starts similarly to the DLT method except that the projective transformation, $\mathbf{P}$, in Eq. (4) now degenerates to a 3×3 homography, $\mathbf{H}$:

$$\mu[x_{\mathrm{p}}\ y_{\mathrm{p}}\ 1]^{\mathrm{T}} = \mathbf{H}\,[\,X_{\mathrm{w}}\ Y_{\mathrm{w}}\ 1\,]^{\mathrm{T}}, \tag{7}$$

where

$$\mathbf{H} = \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{t}], \quad \mathbf{K} = \begin{bmatrix} \alpha & \gamma & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

determines the coordinate transformation between distorted image coordinates, $(x_{\mathrm{d}}, y_{\mathrm{d}})$, and pixel coordinates, $(x_{\mathrm{p}}, y_{\mathrm{p}})$, as shown in Table 1, and $\mathbf{r}_i$ $(i = 1, 2, 3)$ is the ith column vector of the rotation matrix, $\mathbf{R}$. Hence, for each view of the pattern, a homography, $\mathbf{H}$, is estimated from n feature points on the pattern as in Eq. (5) and refined by the Levenberg–Marquardt algorithm. Based on the constraint that the image of the two circular points must lie on the image of the absolute conic, the proof of which can be found elsewhere [17], a linear equation pair is formed as follows:

$$\begin{bmatrix} \mathbf{v_{12}}^{\mathrm{T}} \\ (\mathbf{v_{11}}^{\mathrm{T}} - \mathbf{v_{22}}^{\mathrm{T}})^{\mathrm{T}} \end{bmatrix} \mathbf{b} = 0, \tag{8}$$

where $\mathbf{v_{ij}} = [h_{i1}h_{j1}\ \ h_{i1}h_{j2}+h_{i2}h_{j1}\ \ h_{i2}h_{j2}\ \ h_{i3}h_{j1}+h_{i1}h_{j3}\ \ h_{i3}h_{j2}+h_{i2}h_{j3}\ \ h_{i3}h_{j3}]^{\mathrm{T}}$, $[h_{i1}\,h_{i2}\,h_{i3}]^{\mathrm{T}}$ is the ith column vector of a homography, $\mathbf{H}$, and $\mathbf{b} = [b_{11}\,b_{12}\,b_{22}\,b_{13}\,b_{23}\,b_{33}]^{\mathrm{T}}$ is a 6D vector representation of the symmetric matrix

$$\mathbf{B} = \lambda\,\mathbf{K}^{-T}\,\mathbf{K}^{-1} = \lambda\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{bmatrix} = \lambda\begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{c_y\gamma - c_x\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(c_y\gamma - c_x\beta)}{\alpha^2\beta^2} - \frac{c_y}{\beta^2} \\ \frac{c_y\gamma - c_x\beta}{\alpha^2\beta} & -\frac{\gamma(c_y\gamma - c_x\beta)}{\alpha^2\beta^2} - \frac{c_y}{\beta^2} & \frac{(c_y\gamma - c_x\beta)^2}{\alpha^2\beta^2} + \frac{c_y^2}{\beta^2} + 1 \end{bmatrix}, \quad (9)$$

where λ is an arbitrary scale factor. Given $m \geq 3$ views of the pattern, *i.e.*, the m homographies obtained above, an over-determined system of m pairs of Eq. (8) can be established to obtain the vector, **b**, from which the intrinsic parameters can be extracted. The extrinsic parameters with respect to each orientation of the pattern plane are computed by Eq. (6). Finally, the Levenberg–Marquardt algorithm optimizes the result while taking into account the radial distortions, $k_1^{(Z)}$, $k_2^{(Z)}$.

2.3 Evaluation of calibration accuracy

In our experiments, a camera is calibrated by Tsai, Heikkilä and Zhang's methods using a set of training data,[1] and then validated with a neutral test set covering a wide distance range. Some well developed, freely available calibration toolboxes are used as implementations of the three algorithms in order to achieve a fair comparison: Reg Willson's code[2] for Tsai's algorithm, Janne Heikkilä's toolbox[3] for Heikkilä's algorithm, and Jean-Yves Bouguet's toolbox[4] for Zhang's algorithm. For evaluating both training and testing accuracies, four of the most frequently used methods [6] were adopted based on their applicability to the single camera calibration case.

The *error of distorted pixel coordinates*, E_d, is measured by computing the discrepancy between estimated pixel coordinates, $(\hat{x}_{\mathrm{p}i},\ \hat{y}_{\mathrm{p}i})$, projected from measured world coordinates by the camera model with lens distortions, and observed pixel coordinates, $(x_{\mathrm{p}i},\ y_{\mathrm{p}i})$, obtained from captured images:

$$E_\mathrm{d} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(\hat{x}_{\mathrm{p}i} - x_{\mathrm{p}i})^2 + (\hat{y}_{\mathrm{p}i} - y_{\mathrm{p}i})^2}, \quad (10)$$

where n is the number of feature points.

The *error of undistorted pixel coordinates*, E_u, is measured by computing the discrepancy between estimated undistorted pixel coordinates, $(\hat{x}_{\mathrm{up}i},\ \hat{y}_{\mathrm{up}i})$, projected from measured world coordinates without lens distortions, and observed undistorted pixel coordinates, $(x_{\mathrm{up}i},\ y_{\mathrm{up}i})$, computed by removing distortions from observed pixel coordinates:

$$E_\mathrm{u} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(\hat{x}_{\mathrm{up}i} - x_{\mathrm{up}i})^2 + (\hat{y}_{\mathrm{up}i} - y_{\mathrm{up}i})^2}. \quad (11)$$

The *distance with respect to the optical ray*, E_o, is measured between 3D points in camera coordinates, $(X_{ci},\ Y_{ci},\ Z_{ci})$, and the optical rays back-projected from the corresponding undistorted image points on the camera image plane, $(x_{\mathrm{u}i},\ y_{\mathrm{u}i})$. For Tsai and Heikkilä's models, E_o is expressed as:

$$E_\mathrm{o}^{(\mathrm{T\_H})} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(X_{ci} - x_{\mathrm{u}i}\cdot t)^2 + (Y_{ci} - y_{\mathrm{u}i}\cdot t)^2 + (Z_{ci} - f\cdot t)^2},$$
$$t = (X_{ci}x_{\mathrm{u}i} + Y_{ci}y_{\mathrm{u}i} + Z_{ci}f)\,/\,\left(x_{\mathrm{u}i}^2 + y_{\mathrm{u}i}^2 + f^2\right); \quad (12)$$

and for Zhang's model,

$$E_\mathrm{o}^{(\mathrm{Z})} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(X_{ci} - x_{\mathrm{u}i}\cdot t)^2 + (Y_{ci} - y_{\mathrm{u}i}\cdot t)^2 + (Z_{ci} - t)^2},$$
$$t = (X_{ci}x_{\mathrm{u}i} + Y_{ci}y_{\mathrm{u}i} + Z_{ci})\,/\,\left(x_{\mathrm{u}i}^2 + y_{\mathrm{u}i}^2 + 1\right). \quad (13)$$

These three measurements are intuitive but sensitive to digital image resolution, camera field-of-view, and object-to-camera distance. *Normalized calibration error (NCE)*, proposed by Weng et al. [9] overcomes this sensitivity by normalizing the discrepancy between estimated and observed 3D points with respect to the area each back-projected pixel covers at a given distance from the camera. (See Fig. 1.) The NCE is calculated as follows:

$$E_n = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{(\hat{X}_{ci} - X_{ci})^2 + (\hat{Y}_{ci} - Y_{ci})^2}{Z_{ci}^2(\alpha^{-2} + \beta^{-2})/12}\right]^{1/2}, \quad (14)$$

where $(\hat{X}_{ci},\ \hat{Y}_{ci},\ Z_{ci})$ represent 3D camera coordinates as estimated by back-projection from 2D pixel coordinates to depth Z_{ci} and $(X_{ci},\ Y_{ci},\ Z_{ci})$ represent observed 3D camera coordinates computed from measured 3D world coordinates. In Tsai and Heikkilä's methods, the values of α and β can be calculated using Eq. (1).

[1] In real data experiments, training sets are acquired separately for each of the three methods according to their varying requirements.

[2] http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html

[3] http://www.ee.oulu.fi/~jth/calibr/

[4] http://www.vision.caltech.edu/bouguetj/calib_doc

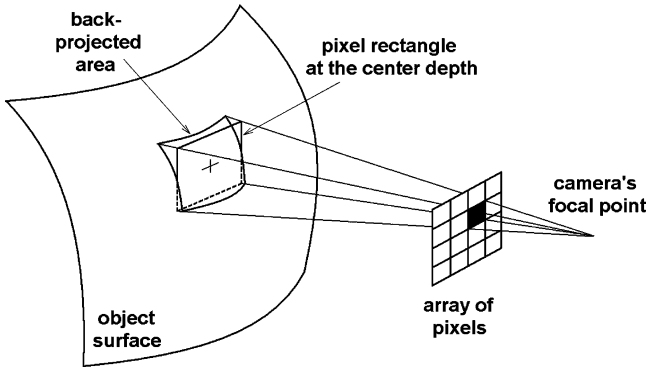


Fig. 1 Back-projection of a pixel to the object surface. The back-projected area on the object surface is represented by a pixel rectangle at the center depth. The error between the back-projected area and the pixel rectangle is measured to assess the calibration accuracy. Based on Fig. 6 of Weng et al. [9]

3 Computer simulations

Our simulated camera had the following properties, chosen based on empirical data: center-to-center distances between adjacent pixels in x and y directions of $d_x = 1/D_x = 0.016$ mm and $d_y = 1/D_y = 0.010$ mm, an image scale factor $s_x = 1.5$, and an effective focal length $f = 8$ mm, resulting in pixel focal lengths of $\alpha = 750$ pixels and $\beta = 800$ pixels. A second-order radial distortion was simulated with the coefficients $k_1^{(Z)} = -0.32$ mm^{-2} and $k_2^{(Z)} = 0$ in Zhang's model, or equivalently, $k_1^{(T)} = k_1^{(H)} = 0.005709$ mm^{-2} and $k_2^{(H)} = p_1^{(H)} = p_2^{(H)} = 0$ in Tsai and Heikkilä's models. The skew parameter, γ, was set to 0. The image resolution was 512×512 pixels with the image center at $(c_x, c_y) = (264, 280)$ pixels.

The training points of all the three methods, covering 30–55 cm from the simulated camera, were obtained from the grid corners of 20 cm $\times$ 20 cm simulated checkerboard patterns, placed at 16 different orientations in front of the virtual camera at a 45° angle with respect to the image plane.[5] The number of grid corners contained in a simulated pattern varies according to the requirements of each experiment.

The 4,108 testing points, covering a distance of 10–300 cm, were generated by sampling a 3 m $\times$ 3 m $\times$ 3 m cubic space at intervals of 10 cm in all three dimensions and keeping points visible to a camera located at the center of one face of the cube.

3.1 Effect of noise on calibration accuracy

We simulated 16 views of a 10×10 checkerboard pattern to generate 1,600 training points. Different levels of Gaussian noise were added to study its effect on calibration accuracy.

3.1.1 Calibration accuracy vs. pixel coordinate noise

Figure 2 shows the decrease in training and testing accuracies of all tested methods as pixel noise increases.[6] However, Zhang's algorithm was found to be more sensitive to pixel coordinate noise and hence more dependent on high-accuracy calibration feature detection than either Tsai or Heikkilä's.

3.1.2 Calibration accuracy vs. world coordinate noise

Figure 3 illustrates the expected decrease of calibration accuracy as world coordinate noise increases. Zhang's calibration error was, again, the highest for the same noise range. While these results might at first seem discouraging for Zhang's method, it is important to note that unlike the other algorithms, which require absolute coordinate measurement with respect to a fixed world reference, Zhang's relative world coordinate measurement allows for a minimal equipment requirement, which, in practice, can simply be a laser printed checkerboard pattern on a letter sized sheet. As a result, most setups can easily achieve measurement noise levels of $\sigma < 0.5$ mm, obtaining a reasonably high calibration accuracy, as shown in the last column of Fig. 3. In contrast, Tsai and Heikkilä's world coordinate measurements are inherently prone to higher noise; the fact that their testing errors increased significantly when $\sigma > 2$ mm poses a strong constraint on real-world setups for accurate measurement. Although largely similar, Heikkilä's algorithm performed better than Tsai's at higher noise level, but slightly worse at lower noise levels. This may be due to Heikkilä's use of fixed empirical values to initialize the linear estimation of intrinsic parameters, which, while more robust to measurement errors, failed to exploit the potential of accurate measurements.

3.2 Effect of training data quantity on accuracy

As the effect of the number of orientations and the angle of the pattern plane on calibration accuracy has been studied by Zhang [2],[5] this section explores the effect of the number of feature points per pattern on calibration accuracy. In the absence of noise, a small number of training points can yield 100% accuracy. As some existing corner detection algorithms claim an accuracy of 0.1 pixels, Gaussian noise of zero mean and $\sigma = 0.1$ pixels was added to the pixel coordinates of training data. The average results of 10 trials are illustrated in Fig. 4.

As each checkerboard pattern was viewed at 16 different orientations, a pattern of 3×3 grid corners could produce 144 training points, sufficient for Tsai's algorithm to achieve reasonable accuracy; however, the calibration error stabilized further when more than 256 training points were

[5] The number of orientations and the angle of the pattern plane were chosen according to Zhang [2], in which the best performance was reported with more than 10 orientations and an angle near 45°.

[6] All four evaluation methods described in Sect. 2.3 were used and results demonstrated very similar trends. Due to limited space, only the normalized calibration error (NCE) is plotted.

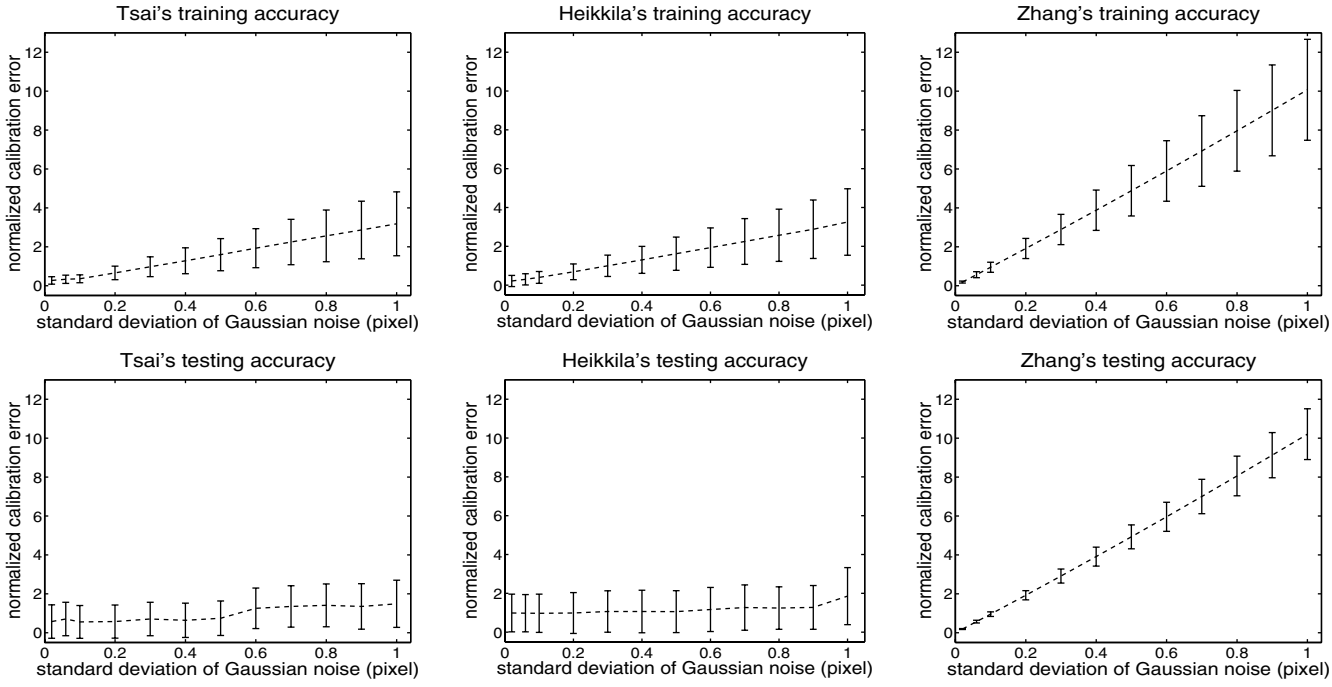


Fig. 2 Effect of pixel coordinate noise on calibration accuracy. Gaussian noise of $\mu = 0$, σ of 0.02–1.0 pixels added to training data pixel coordinates. No noise added to test data. Note that zero testing error was achieved when adding zero noise

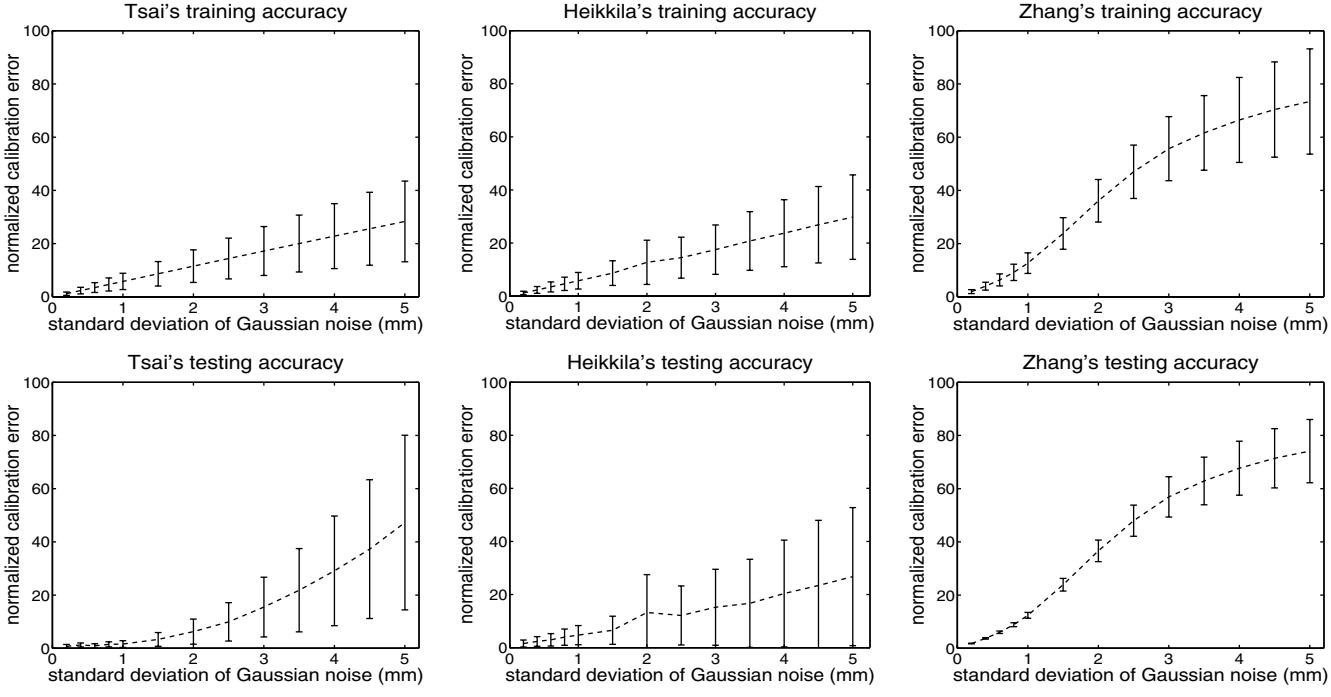


Fig. 3 Effect of world coordinate noise on calibration accuracy. Gaussian noise of $\mu = 0$, σ of 0.2–5.0 mm added to 3D world coordinates of Tsai and Heikkilä's training data and 2D world coordinates of Zhang's training data, as Zhang's algorithm assumes all feature points fall on the $Z_w = 0$ plane. No noise added to test data

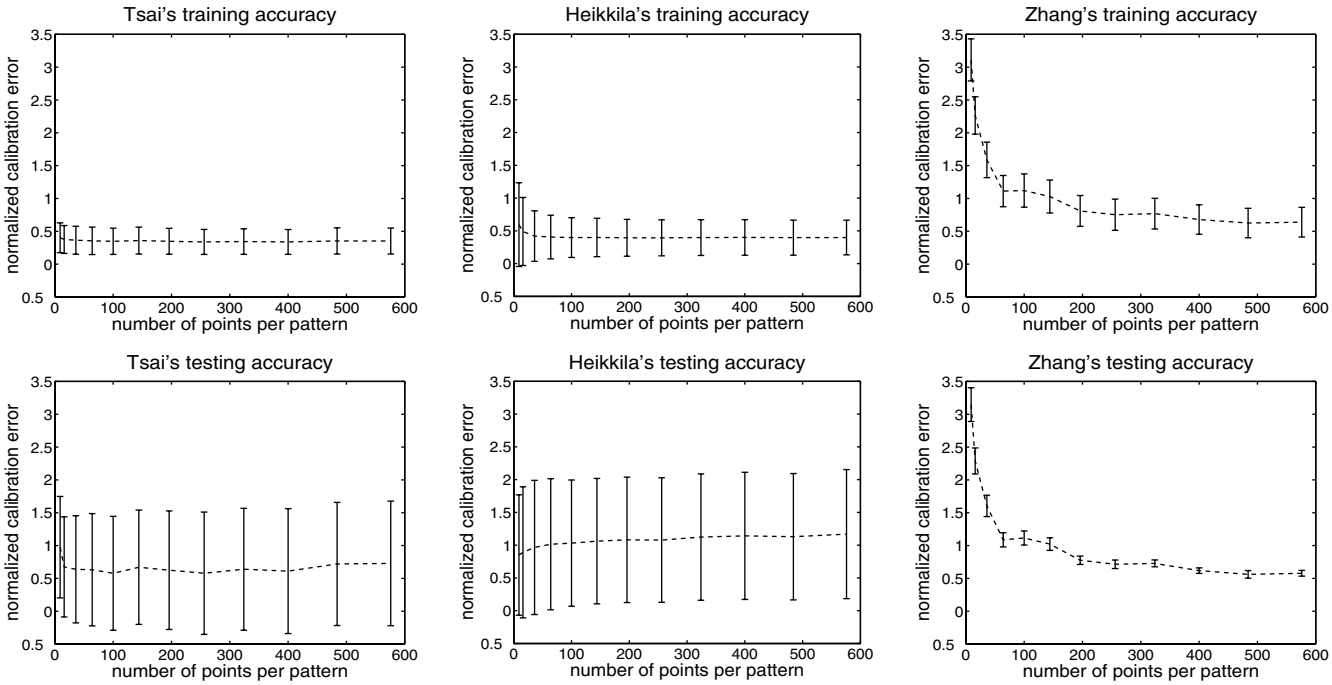


Fig. 4 Effect of training data quantity on calibration accuracy. Training data generated from 16 views of checkerboard patterns containing between 3×3 and 24×24 grid corners. Gaussian noise of $\mu = 0, \sigma = 0.1$ pixels added to training data pixel coordinates. No noise added to test data

used. Although more robust with limited training data quantities, Heikkilä's results demonstrated a slightly inferior performance to Tsai's, which might be explained by the low noise level in training data as suggested in the previous section. Zhang's calibration error was again higher than those of the other two methods with small training sets, as the former is more sensitive to noise. However, increasing the number of training points per pattern alleviates this sensitivity, resulting in similar accuracies to Tsai's algorithm when employing more than 200 training points per pattern. We also notice in Fig. 4 that testing errors exhibit higher standard deviation with Tsai and Heikkilä's algorithms than with Zhang's. This is likely due to the fact that the former two treated each feature point independently whereas the latter took advantage of the coplanar constraint between feature points of each view, thus compensating for the inconsistencies of noisy training points.

3.3 Effect of distortion model on calibration accuracy

Radial and decentering distortions [23] are the most common distortions modeled in camera calibration and can be expressed as follows:

$$\begin{bmatrix} x_\mathrm{d} \\ y_\mathrm{d} \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots) \begin{bmatrix} x_\mathrm{u} \\ y_\mathrm{u} \end{bmatrix} + \begin{bmatrix} 2p_1 x_\mathrm{u} y_\mathrm{u} + p_2\left(r^2 + 2x_\mathrm{u}^2\right) \\ p_1\left(r^2 + 2y_\mathrm{u}^2\right) + 2p_2 x_\mathrm{u} y_\mathrm{u} \end{bmatrix} \quad (15)$$

where k_i $(i = 1, 2, \ldots)$ and p_1, p_2 are radial and decentering distortion coefficients, and $r = \sqrt{x_\mathrm{u}^2 + y_\mathrm{u}^2}$.

In this experiment, five types of cameras were simulated, each corresponding to a different distortion characteristic that consists of the first n low-order radial distortion terms with or without the two decentering terms, Rn (n = 1, 2) and RnD2 (n = 1, 2, 3). The simulated coefficients, listed in Table 2, were chosen from empirical data. All the remaining camera parameters were the same as previously described. Each of the five simulated cameras was calibrated by Zhang's algorithm combined, in turn, with each of the five distortion models, Rn (n = 1, 2) and RnD2 (n = 1, 2, 3), with skew parameter set to zero.

Figure 5 shows, for each simulated camera, the calibration error versus the distortion models used for calibration on a large low-noise training set. The results indicate that high calibration accuracy is obtained provided that the distortion model assumed in calibration includes all the distortion components of the camera, although the

Table 2 Distortion coefficients of simulated cameras

Distortion Coefficients	R1	R2	R1D2	R2D2	R3D2
k_1 (mm^{-2})	−0.3	−0.3	−0.3	−0.3	−0.3
k_2 (mm^{-4})	0	0.15	0	0.15	0.15
k_3 (mm^{-6})	0	0	0	0	0.1
p_1 (mm^{-2})	0	0	0.02	0.02	0.02
p_2 (mm^{-2})	0	0	0.015	0.015	0.015

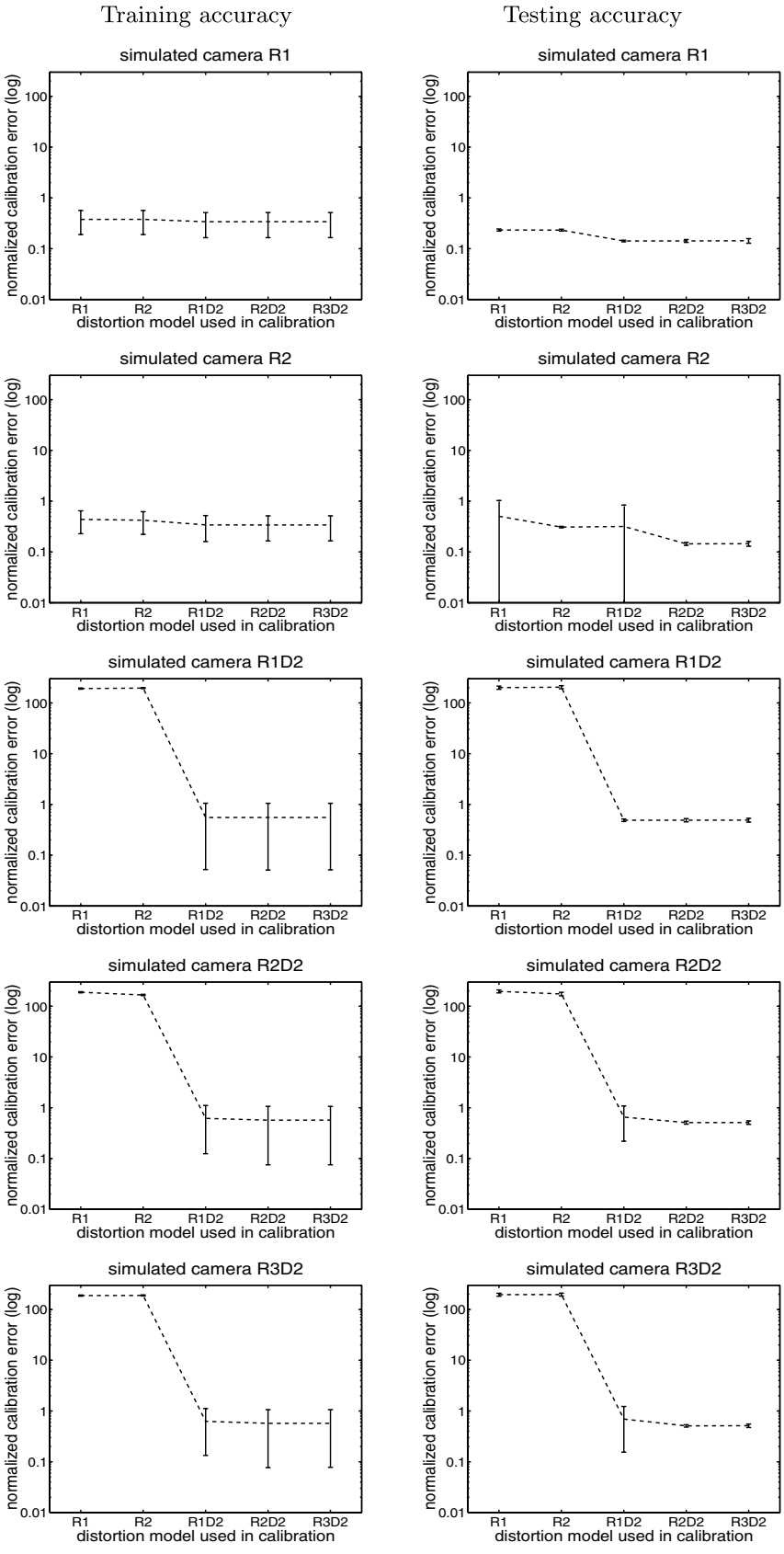


Fig. 5 Calibration accuracy vs. distortion model. Trained from 16 views of 20×20 pattern with Gaussian noise of $\mu = 0, \sigma = 0.1$ pixels added to pixel coordinates. No noise added to test data. Logarithmic scale used for y-axis. Measured with Zhang's algorithm

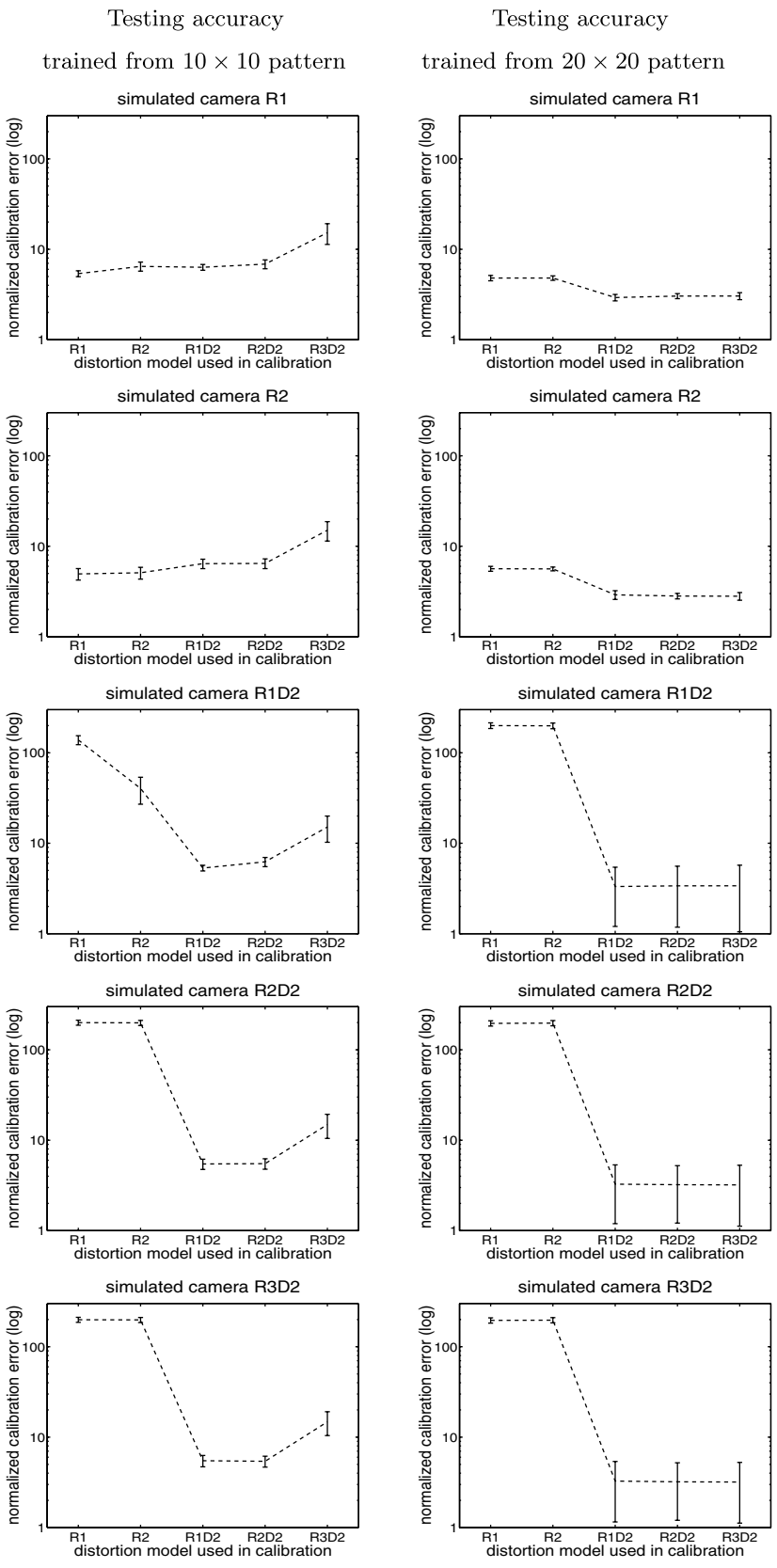


Fig. 6 Testing accuracy vs. distortion model. Trained from 16 views of 10×10 (left) and 20×20 (right) patterns with Gaussian noise of $\mu = 0$, $\sigma = 0.5$ pixels or 0.5 mm added to pixel and world coordinates. No noise added to test data. Logarithmic scale used for y-axis. Measured with Zhang's algorithm

Table 3 Simulated camera R2D2 and calibration results using different distortion models

Camera parameters	Simulated R2D2	Distortion models used in calibration				
		R1	R2	R1D2	R2D2	R3D2
α (pixels)	750	754.36	754.32	749.46	749.98	749.98
β (pixels)	800	802.98	802.88	799.51	799.98	799.98
c_x (pixels)	264	220.06	224.67	263.80	263.82	263.82
c_y (pixels)	280	218.50	217.23	279.94	279.95	279.95
k_1 (mm^{-2})	−0.3	−0.1422	−0.1019	−0.2812	−0.3005	−0.3001
k_2 (mm^{-4})	0.15	–	−0.1460	–	0.1538	0.1472
k_3 (mm^{-6})	0	–	–	–	–	0.03055
p_1 (mm^{-2})	0.02	–	–	0.01995	0.01996	0.01996
p_2 (mm^{-2})	0.015	–	–	0.01497	0.01497	0.01497
Normalized calibration error	training	188.25	166.44	0.6179	0.5699	0.5701
	testing	196.14	174.20	0.6537	0.5102	0.5113

sixth-order radial term does not benefit accuracy. Moreover, adding higher order radial terms affects the estimation of lower order terms, as can be observed in Table 3. This correlation between radial distortion components may, unfortunately, degrade calibration performance when only a limited amount of noisy training data is available. Figure 6 illustrates the same experiment using noisy training data. The addition of higher order radial terms yielded a higher error, especially with small training sets, as shown in the left column. Nonetheless, including the two decentering distortion components generally guaranteed a high calibration accuracy for a camera with unknown lens distortions.

4 Real data experiments

The real data experiments were carried out in our Shared Reality Environment (SRE), a laboratory space equipped with three vertical projection screens, each approximately 2.5 m × 1.8 m and raised about 0.7 m above the ground, semi-enclosing a space of 5.4 m^2 as shown in Fig. 7. A 3Com U.S. Robotics BigPicture Camera with fixed focal length was placed along one screen at a height of 0.5 m, facing the other two adjacent screens. The image resolution was 640 × 480 pixels. To investigate the influence of experimental setup, two configurations, the *casual setup* and the *elaborate setup*, were studied.

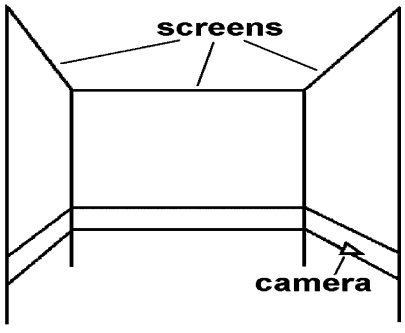


Fig. 7 A node of the Shared Reality Environment

4.1 Casual setup

The training data for Tsai and Heikkilä's calibration was obtained from 600 grid corners of checkerboard patterns of 8×8 cm squares, projected onto the two visible screens, as in Fig. 8a.[7] Assuming that the projectors were accurately calibrated and the patterns were projected as rectangular shapes, the 3D world coordinates of the four pattern corners on each screen were measured according to a fixed world reference system referred to as SRE coordinates, and the remaining points were interpolated. Due to the calibration error between projectors and screens, the limited image resolution of the projected pattern, and the non-rigid material of the screens, verifying with a few interior points indicated an average interpolation error of 4.1 mm, equivalent to approximately 0.28% of the pattern size. This data set covered a distance of 200–325 cm from the camera and will be referred to as Screen Data in the following text.

The training data for Zhang's calibration was generated by printing checkerboard patterns of 8 × 6, 12 × 9, 15 × 14, and 20 × 20 grid corners onto letter sized sheets. Each was attached to a rigid card[8] and viewed at 16 different orientations at roughly 45° with respect to the camera image plane, as explained in Sect. 3 and demonstrated in Fig. 8b. This produced four data sets of 768–6400 points. The 2D relative world coordinates of these points were measured with respect to one of the four corners of each pattern, i.e. the origin of a changing world reference system. The four corners were measured first and the interior points interpolated due to the regularity of the printed pattern. As our ruler was accurate only to 1 mm, we assume a maximum measurement error of 0.5 mm, approximately 0.29% of the pattern size. These

[7] Due to the difficulty of achieving 3D measurements of low errors on an arbitrarily positioned small checkerboard pattern, we did not use 16 views of a hand-held checkerboard pattern for Tsai and Heikkilä's training data collection as in the simulation experiments, Sect. 3. Instead, we used large checkerboard patterns projected on big screens, which is likely to produce highly accurate measurements to the advantage of Tsai and Heikkilä's calibration.

[8] According to Zhang's study [2], the effect of systematic non-planarity can be ignored in our experiments.

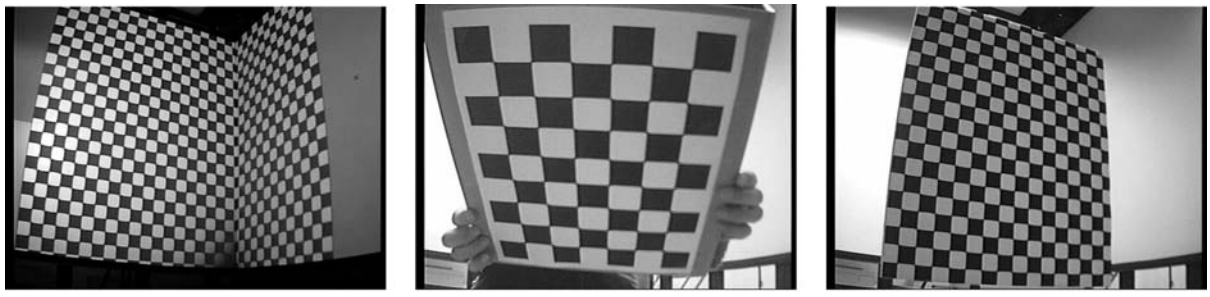


Fig. 8 A demonstration of *casual setup* for generating **a** Tsai and Heikkilä's training data, **b** Zhang's training data, and **c** test data

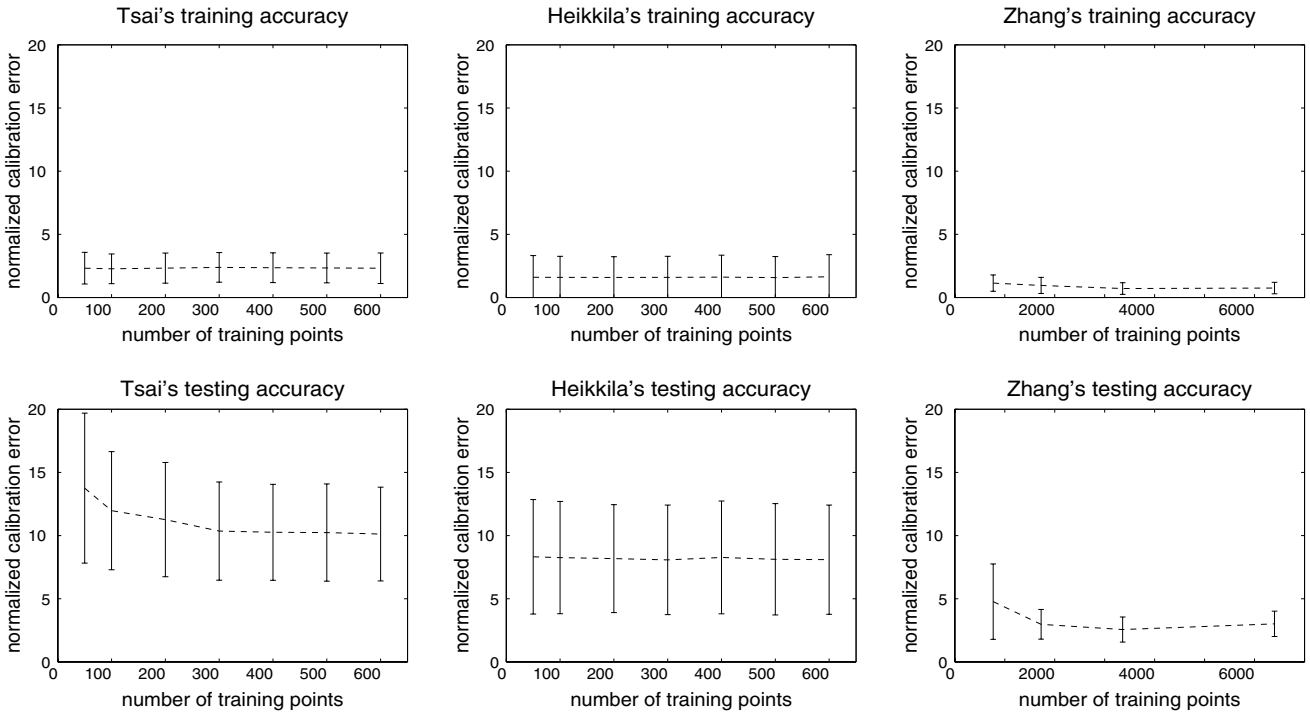


Fig. 9 Effect of training data quantity on calibration accuracy in *casual setup*. Admittedly, more training points were used for Zhang's calibration than for Tsai and Heikkilä's. However, as indicated by the simulation results in Fig. 4, there was no accuracy improvement in Tsai and Heikkilä's results beyong 256 training samples. This is also evident in the present figure and Fig. 14 of our *elaborate setup*

four data sets each covered a distance of 25–55 cm from the camera and will be referred to as Board48, Board108, Board210, and Board400 Data respectively.

The test set for all the three algorithms was created by moving a wooden board bearing a 100 cm × 85 cm printed pattern of 17 × 15 checks along a specially constructed rail at different locations within the SRE space, as pictured in Fig. 8c, to provide 1,872 accurately measured points in the SRE coordinates. This data set covered a distance of 100–265 cm and will be referred to as Rail Data.

4.1.1 Effect of training data quantity

Tsai and Heikkilä's algorithms were trained using between 50 and 600 points, selected from Screen Data to be evenly distributed across the screens, and then tested on Rail Data. The average results of 10 trials for each quantity of training data are shown in Fig. 9. With Tsai's algorithm, no significant improvements in testing accuracy were observed beyond 300 training samples. Similar to the simulation results, Heikkilä's algorithm produced better results than Tsai's with small training data quantities. The overall lower testing errors of Heikkilä's method than Tsai's suggested the presence of high noise in the training data, as discussed in Sect. 3.1.

Zhang's algorithm was trained separately on Board48, Board108, Board210, and Board400 Data, and tested on Rail Data. As Zhang's extrinsic parameters were calibrated with respect to a corner point of the calibration pattern, whose position in the SRE coordinates varied over different views,

Table 4 The best calibration results obtained in *casual setup*

Camera parameters	Tsai[a]	Heikkilä[a]	Zhang[b]
Intrinsic			
c_x (pixels)	338.56	319.55	332.39
c_y (pixels)	240.34	263.12	268.01
f (mm)	6.9333	6.7617	
s_x	1.5575	1.5504	
d_x (mm)	0.016 (given)		
d_y (mm)	0.010 (given)		
D_x (mm^{-1})		62.5 (given)	
D_y (mm^{-1})		100.0 (given)	
α (pixels)			675.89
β (pixels)			695.15
γ			0.000533
$k_1^{(\text{T})}$ (mm^{-2})	0.007121		
$k_1^{(\text{H})}$ (mm^{-2})		0.006477	
$k_2^{(\text{H})}$ (mm^{-4})		0.000333	
$k_1^{(\text{Z})}$ (mm^{-2})			−0.3230
$k_2^{(\text{Z})}$ (mm^{-4})			0.08667
$p_1^{(\text{H})}$ (mm^{-2})		0.000627	
$p_2^{(\text{H})}$ (mm^{-2})		0.001838	
Extrinsic			
$\mathbf{t}$ (mm)	$\begin{bmatrix} -1982.9 \\ -1455.5 \\ 2324.9 \end{bmatrix}$	$\begin{bmatrix} -1910.2 \\ -1535.9 \\ 2256.4 \end{bmatrix}$	$\begin{bmatrix} -1963.5 \\ -1560.0 \\ 2269.7 \end{bmatrix}$
$\mathbf{r}_\theta$ (°)	$\begin{bmatrix} 153.18 \\ 19.44 \\ 91.25 \end{bmatrix}$	$\begin{bmatrix} 154.73 \\ 17.69 \\ 91.81 \end{bmatrix}$	$\begin{bmatrix} 153.07 \\ 17.33 \\ 91.45 \end{bmatrix}$

[a]Trained on Screen Data
[b]Trained on Board210 Data with extrinsic recalibration

Table 5 Comparing calibration accuracies assuming skewness $\gamma \neq 0$ and $\gamma = 0$

Accuracy evaluation	Accuracy			
	Training		Testing	
	$\gamma \neq 0$	$\gamma = 0$	$\gamma \neq 0$	$\gamma = 0$
2D distorted error (pixels)	0.2776	0.2782	1.0028	0.8959
2D undistorted error (pixels)	0.2898	0.2905	1.0470	0.9395
Distance from optical ray (mm)	0.1648	0.1653	2.7689	2.5153
Normalized calibration error	0.7099	0.7118	2.5602	2.3076

while the Rail Data were measured in the fixed SRE coordinates, the extrinsic parameters needed to be recalibrated to the same coordinate system. The training data for this extrinsic recalibration was generated by placing a printed 12 × 15 checkerboard pattern at a location along the rail to produce feature points measured in SRE coordinates. After extrinsic recalibration, the testing accuracies on Rail Data were obtained, as shown in Fig. 9. Both training and testing accuracies increased with the number of training data per pattern with no significant improvement beyond 100 samples per pattern, which is consistent with the simulations. Recalling that the test data (100–265 cm from the camera) was obtained from a much further distance range than the training data (25–55 cm), one might expect large errors in the test results given the need for extrapolation from training data. However, despite this large discrepancy, Zhang's algorithm achieved impressive testing accuracies. This suggests that the parameters calibrated by Zhang's algorithm are scalable to test data over a larger range than that of the training data.

Table 4 lists the best calibration results of each algorithm from Fig. 9, with the corresponding accuracy shown in Table 6, *casual setup*. Under our experimental conditions with the interpolation measurement errors as described previously, Zhang's algorithm outperformed Tsai and Heikkilä's by approximately four and three times, respectively under all evaluation measures.

4.1.2 Effect of distortion model

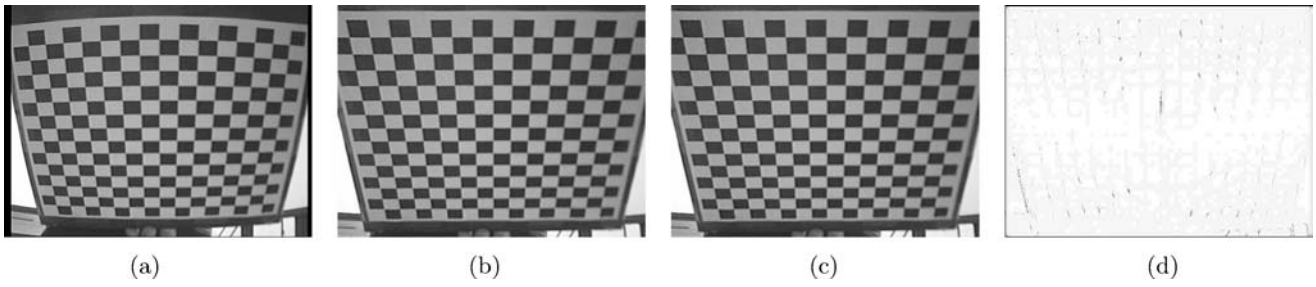
The 3Com U.S. Robotics BigPicture Camera exhibits obvious lens distortions visible in the images of Fig. 8. We studied the effect of distortion model on calibration accuracy.

Skewness. Although included in a linear transformation and not part of the actual distortion model, skewness should nevertheless be considered as a type of distortion. In Zhang's method, the skewness was estimated, as expressed by γ in Table 4, but was essentially zero. For comparison, camera parameters were calibrated by Zhang's algorithm on Board210 Data with γ either estimated or fixed at zero. The calibration accuracies are compared in Table 5, showing no improvement when estimating γ. This result can also be seen in Fig. 10.

Lens distortion. Zhang's algorithm was integrated separately with each of the five distortion models described in Sect. 3.3 to calibrate camera parameters on Board210 Data. Calibration accuracies are displayed in Fig. 11. While all models achieved almost the same training accuracy, those considering decentering distortions performed marginally better in testing. As shown in Fig. 12, all five models successfully recovered the original image from distortions with little difference in the results.

4.2 Elaborate setup

To investigate how much improvement can be realized by increasing the measurement accuracy of training data and by reducing the discrepancy of distance coverage between training and test set, the rail structure for obtaining the test data in Sect. 4.1 was used to generate a total of 3,810 accurately measured feature points in an attempt to cover the volume of the camera's working space within the SRE. From this set, 50–2,000 evenly distributed points, covering 85–245 cm from the camera, were selected as Tsai and Heikkilä's training data and the remaining points, covering the same range, were used as a neutral test set for all



(a) (b) (c) (d)

Fig. 10 Removing distortions from **a** original image using Zhang's model assuming **b** skewness $\gamma \neq 0$ and **c** $\gamma = 0$ with **d** their difference, i.e. $|(b) - (c)|$, which is barely visible. (We use *white* to denote values of zero difference)

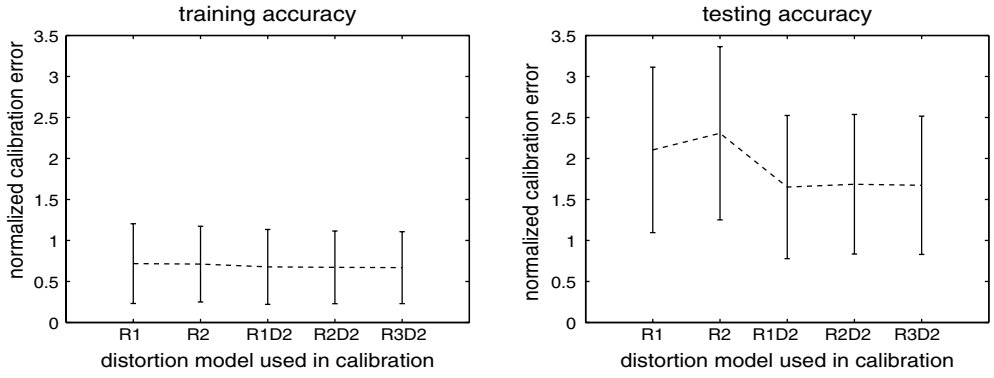
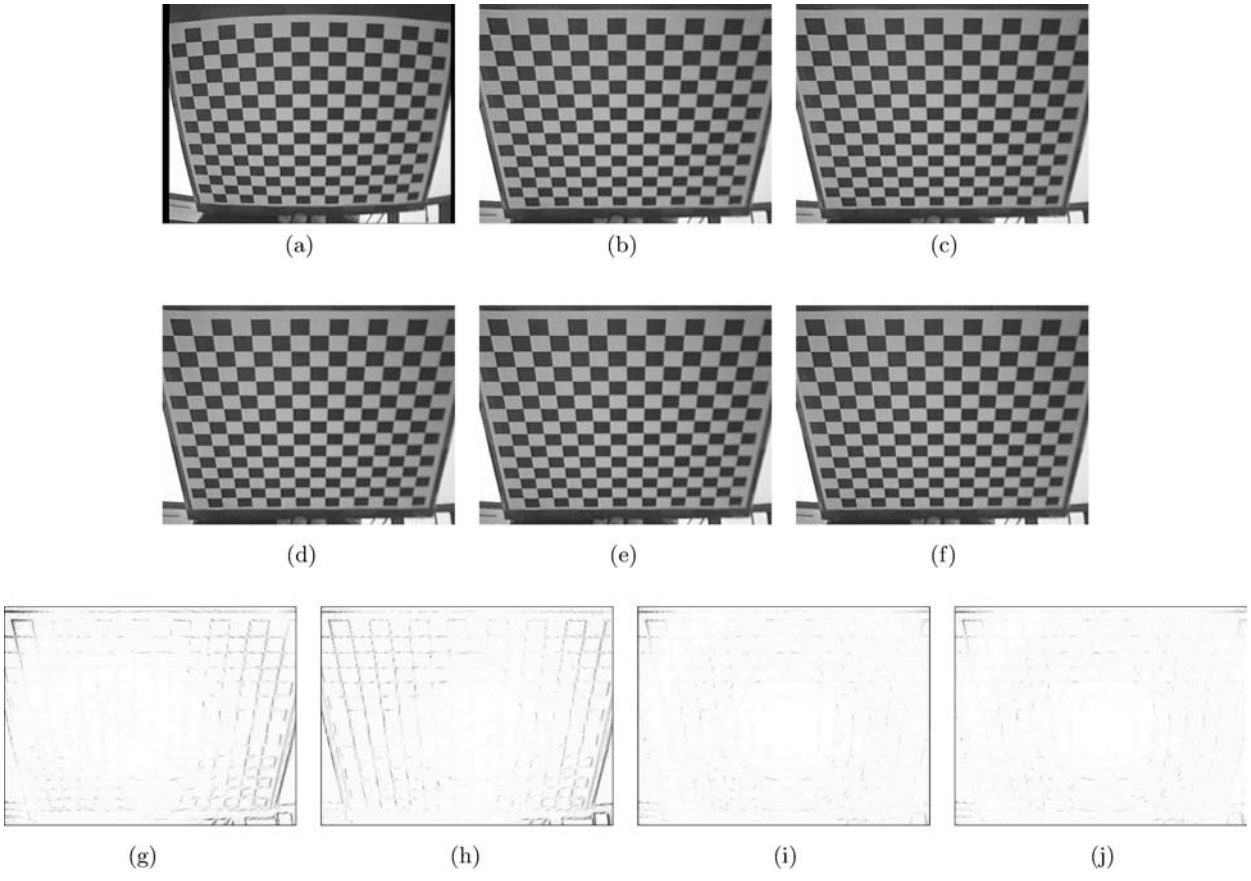


Fig. 11 Calibration accuracy vs. distortion model used in calibration. Trained from 16 views of a 15×14 checkerboard pattern



(a) (b) (c)

(d) (e) (f)

(g) (h) (i) (j)

Fig. 12 Removing lens distortions from **a** original image using distortion model **b** R1, **c** R2, **d** R1D2, **e** R2D2 and **f** R3D2. The differences of **b**, **c**, **d** and **e** with respect to **f** are shown in **g**, **h**, **i**, and **j**, respectively. (We use *white* to denote values of zero difference)

Table 6 Accuracy comparison of Tsai, Heikkilä and Zhang's calibration algorithms

		Accuracy					
		Training			Testing		
Experiment setup	Accuracy evaluation	Tsai	Heikkilä	Zhang	Tsai	Heikkilä	Zhang
Casual	2D distorted error (pixels)	0.8843	0.4104	0.2776	3.9206	3.2311	1.0028
	2D undistorted error (pixels)	0.9410	0.6677	0.2898	4.0865	3.2949	1.0470
	Distance from optical ray (mm)	3.5752	2.4291	0.1648	9.9358	7.7337	2.7689
	Normalized calibration error	2.3150	1.6397	0.7099	10.1184	8.0920	2.5602
Elaborate	2D distorted error (pixels)	0.3498	0.3480	0.1816	0.3445	0.3505	0.6301
	2D undistorted error (pixels)	0.3730	1.0765	0.1907	0.3701	1.1900	0.6856
	Distance from optical ray (mm)	0.9456	2.5730	0.4135	0.9244	2.8083	1.6926
	Normalized calibration error	0.9141	2.6484	0.4667	0.9079	2.9267	1.6767

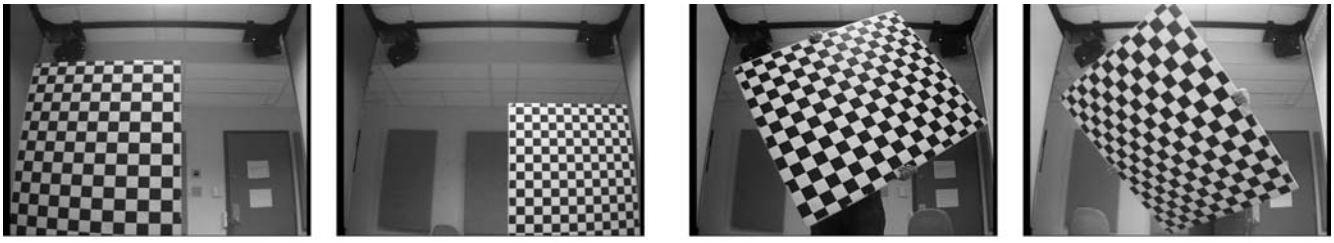


Fig. 13 A demonstration of *elaborate setup* for generating **a** Tsai and Heikkilä's training data and test data, **b** Zhang's training data

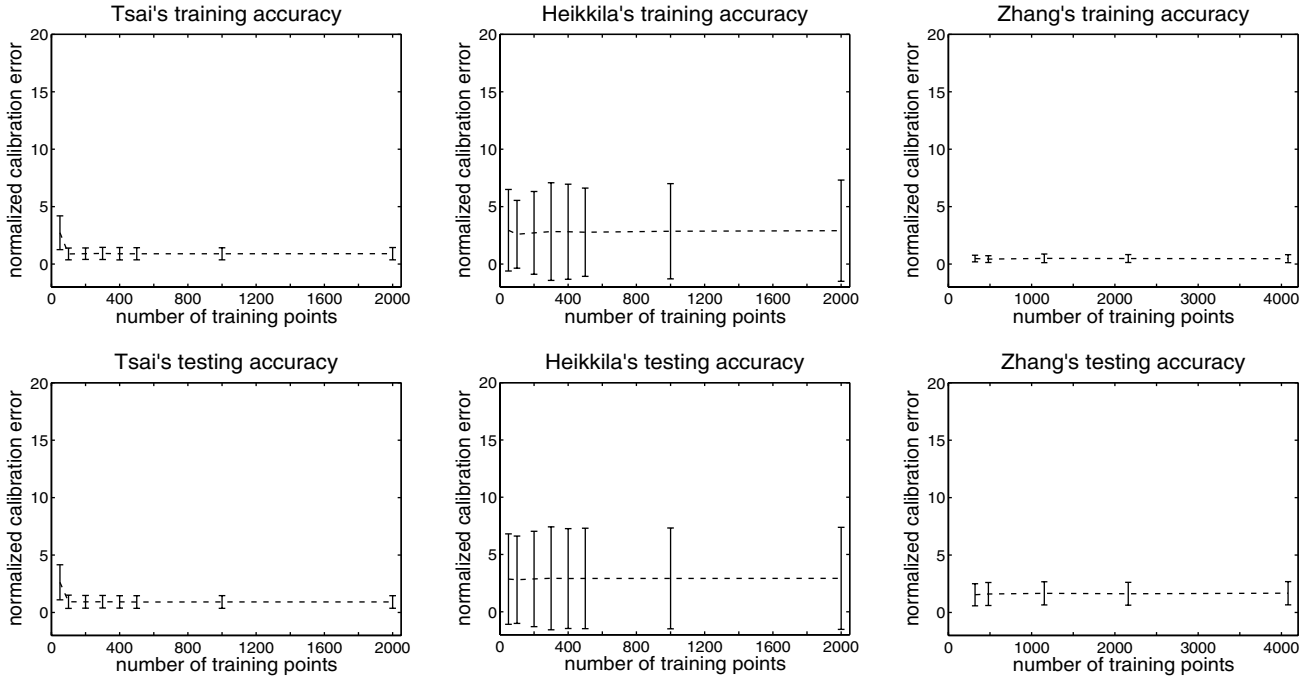


Fig. 14 Effect of training data quantity on calibration accuracy in *elaborate setup*

three algorithms, as demonstrated in Fig. 13a. Zhang's training data was generated in the same manner as described in Sect. 4.1 but replacing the letter sized cardboard pattern with the 100 cm × 85 cm wooden board pattern of the rail structure, producing 16 views of 20–255 planar points covering 95–225 cm from the camera, as demonstrated in Fig. 13b. The extrinsic parameters were then recalibrated with respect to the SRE coordinates by aligning this wooden board pattern with the projection screens at a known location on the rail.

The training and testing accuracy of all three algorithms are shown in Fig. 14 and Table 6, *elaborate setup*. Compared to the best results obtained in the *casual setup*, Tsai and Heikkilä's testing errors decreased by approximately 90%

and 65% while Zhang's error decreased by only 35% as, understandably, there was essentially no increase in training data accuracy of the latter. However, the distance change in training samples from the *casual setup* (25–55 cm) to the *elaborate setup* (95–225 cm) yielded a modest improvement in Zhang's results, as the test data (85–245 cm) was now closer in range to the latter.

5 Conclusion

An empirical study on camera calibration was carried out in the Shared Reality Environment to investigate how such factors as noise level, training data quantity, and distortion model affect calibration accuracy. Three of the most popular and representative methods, developed by Tsai, Heikkilä and Zhang, were chosen for experimentation on both simulated and real data. Four commonly used criteria were applied to evaluate accuracy on separate training and test sets.

Results indicated that the conventional world-reference based approach, exemplified by Tsai's method, can achieve high accuracy when trained on data of low measurement error. However, this requires accurate 3D measurement, typically involving hundreds of samples with respect to a fixed reference system, which is prone to noise, and, as our experiments on actual data confirmed, yields a disappointing NCE of 10.1. After a careful and time-consuming setup and measurement process, we managed to limit NCE to approximately 0.9, indicating that the back-projected 3D space error due to the camera parameters was lower, on average, than the error introduced by quantizing real world 3D coordinates to the level of individual image pixels [9]. However, we note that the effort required to achieve this level of accuracy may well be inordinate for most researchers. Heikkilä's results demonstrated a similar trend despite its slightly greater robustness for small training data quantities and large measurement errors.

In contrast, the planar calibration approach, exemplified by Zhang's method, makes efficient use of world information by taking into account the planar constraint on the calibration pattern, and requires neither a laborious measuring task nor specialized equipment. With a hand-held pattern placed approximately 40 cm from the camera, we obtained an NCE of around 2.6, suggesting an acceptable calibration in which the residual error was almost negligible compared to the pixel quantization error discussed above [9]. Training with a pattern placed closer to the location of the test data yielded improved results, with NCE falling to 1.7.

Although the comparison between the three methods required that the extrinsic parameters obtained by Zhang's method be recalibrated with respect to our fixed SRE coordinates, this process may not be relevant for stereo or multicamera calibration, in which a camera reference system can be used instead of a world reference system. Moreover, the sensitivity of Zhang's algorithm to noise may be overcome by adding training points, simply by printing a checkerboard pattern containing more grid corners. In summary, these results demonstrate the flexibility of the multiplanar approach and its suitability for calibration in dynamic environments.

Our study also included a detailed comparison of distortion models to determine the relative importance of various coefficients given unknown lens distortion. The zero-skewness assumption made in many methods was confirmed as reasonable, at least for the cameras of average quality that we tested, and the second-order term was found sufficient for modeling radial distortion [6]. Estimating the fourth order radial term may be desirable at low noise levels, although including the sixth order term can degrade calibration performance for small noisy training sets. For a camera with unknown lens distortion, adding decentering components, in general, increases the likelihood of accurate calibration.

Acknowledgements This work was supported by a Phase IIIb research grant from the Institute for Robotics and Intelligent Systems (IRIS). The authors thank Jianfeng Yin for help on image capturing and coordinate measurement, Kaleem Siddiqi, James Clark and Stephen Spackman for insightful advice, Don Pavlasek and Jozsef Boka for construction of the measurement assembly. The authors acknowledge the contribution of Reg Willson's calibration code, Janne Heikkilä's calibration toolbox and Jean-Yves Bouguet's calibration toolbox. The authors also wish to thank the journal reviewers for providing numerous helpful suggestions and considerable insights.

References

1. Lavest, J.-M., Viala, M., Dhome, M.: Do we really need an accurate calibration pattern to achieve a reliable camera calibration? In: Proceedings of the European Conference on Computer Vision, pp. 158–174 (1998)
2. Zhang, Z.: A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, http://research.microsoft.com/~zhang/Calib/ (1998)
3. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. IEEE J. Robotics Automat. **3**(4), 323–344 (1987)
4. Heikkilä, J.: Geometric camera calibration using circular control points. IEEE Trans. Pattern Anal. Machine Intell. **22**(10), 1066–1077 (2000)
5. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Machine Intell. **22**(11), 1330–1334 (2000)
6. Salvi, J., Armangué, X., Batlle, J.: A comparative review of camera calibrating methods with accuracy evaluation. Pattern Recognit. **35**, 1617–1635 (2002)
7. Hall, E.L., Tio, J.B.K., McPherson, C.A., Sadjadi, F.A.: Measuring curved surfaces for robot vision. Computer **15**(12), 42–54 (1982)
8. Faugeras, O., Toscani, G.: The calibration problem for stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 15–20 (1986)
9. Weng, J., Cohen, P., Herniou, M.: Camera calibration with distortion models and accuracy evaluation. IEEE Trans. Pattern Anal. Machine Intell. **14**(10), 965–980 (1992)
10. Kanade, T., Rander, P., Narayanan, P.J.: Virtualized reality: constructing virtual worlds from real scenes. IEEE Multimedia, Immersive Telepresence **4**(1), 34–47 (1997)

11. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., Fuchs, H.: The office of the future: a unified approach to image-based modeling and spatially immersive displays. In: SIGGRAPH 98 Proceedings of the 25th Annual Conference Computer Graphics and Interactive Techniques, pp. 179–188 (1998)
12. Fusiello, A.: Uncalibrated euclidean reconstruction: a review. Image Vis. Comput. **18**, 555–563 (2000)
13. Chatterjee, C., Roychowdhury, V.P.: Algorithms for coplanar camera calibration. Machine Vis. Appl. **12**(2), 84–97 (2000)
14. Caprile, B., Torre, V.: Using vanishing points for camera calibration. Int. J. Comput. Vis. **4**(2), 127–140 (1990)
15. Wei, G., Ma, S.: A complete two-plane camera calibration method and experimental comparisons. In: Proceedings of the 4th International Conference on Computer Vision, pp. 439–446 (1993)
16. Faugeras, O., Luong, T., Maybank, S.: Camera self-calibration: theory and experiments. In: Proceedings of the European Conference on Computer Vision, pp. 321–334 (1992)
17. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
18. Bougnoux, S.: From projective to euclidean space under any practical situation, a criticism of self-calibration. In: Proceedings of the 6th International Conference on Computer Vision, pp. 790–796 (1998)
19. Triggs, B.: Autocalibration from planar scenes. In: European Conference on Computer Vision, pp. 89–105 (1998)
20. Sturm, P., Maybank, S.: On plane-based camera calibration: a general algorithm, singularities, applications. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 432–437 (1999)
21. Liebowitz, D., Zisserman, A.: Metric rectification for perspective images of planes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 482–488 (1998)
22. Zhang, Z.: Camera calibration with one-dimensional objects. IEEE Trans. Pattern Anal. Machine Intell. **26**(7), 892–899 (2004)
23. Slama, C.C. (ed): Manual of photogrammetry. American Society of Photogrammetry, 4th edn. Falls Church, VA (1980)
24. More, J.: The Levenberg–Marquardt algorithm, implementation, and theory. In: Watson, G.A. (ed.) Numerical Analysis. Springer-Verlag, New York (1977)

Wei Sun received the B.Eng. degree in Electronic Engineering from Shanghai Jiao Tong University in 1995 and the M.Sc. degree in Computer Science from Fudan University in 1998. At present, she is a Ph.D. candidate working at Centre for Intelligent Machines, Department of Eletrical and Computer Engineering of McGill University. Her research interests include image and video processing, computer vision, machine learning and computer graphics.

Jeremy R. Cooperstock (Ph.D. Toronto, 1996) is a professor of Electrical and Computer Engineering, a member of the Centre for Intelligent Machines, and a founding member of the Centre for Interdisciplinary Research in Music, Media and Technology at McGill University. Cooperstock is a member of the ACM and chairs the AES Technical Committee on Network Audio Systems. Cooperstock's research interests focus on computer mediation to facilitate high-fidelity human communication and the underlying technologies that support this goal.