



# Four points: one-pass geometrical camera calibration algorithm

H. K. Ardakani<sup>1</sup> · A. Mousavinia<sup>1</sup> · Farzad Safaei<sup>2</sup>

Published online: 23 January 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

Conventional geometrical camera calibration algorithms are usually based on running some iterative algorithms on test images obtained carefully from reference objects with precisely known pattern. Providing these test images and running the iterative algorithms are often time-consuming and sometimes costly. In addition, they are usually very sensitive to image distortions. To overcome these problems, an efficient and practical camera calibration method using a single rectangular reference object is proposed. The reference object can be as simple as an A4-size paper placed on a table. Using the coordinate of four corner points of reference image, generate eight equations. This paper first describes an analytical method to solve the equations and then provides a step-by-step algorithm. The proposed algorithm is evaluated using simulated images generated with both Autodesk 3ds Max software and Microsoft Camera Calibration data set. The results show that the accuracy of the proposed method is very close to the best ones available, while its sensitivity to distortion and computational load is the least. In addition, the required reference object is the simplest one.

**Keywords** Camera calibration · Single image · Machine vision · Camera parameters

## 1 Introduction

Camera calibration is often used as an essential early stage in computer vision to derive metric information from 2-D images. The task of camera calibration is to estimate the parameters that govern the relationship between the 2-D image perceived by a camera and the 3-D information of the photographed object [1]. Camera *intrinsic parameters* describe the camera model, and *extrinsic parameters* define the relative location and orientation of the camera in the real world. Camera calibration is an indispensable step in a wide spectrum of applications, such as metrology, video surveillance, augmented reality presentation, virtual content insertion and 3-D reconstruction to provide the geometric mapping between 2-D image positions and 3-D real-world

coordinates [2–4]. Several methods have been proposed to accomplish this task.

Camera calibration methods that use collinear points on a 1-D object with rotation, planar motion, general motion and non-collinear points under special motion require two or more images of the reference object, while it would be more practical if the calibration method required only a single image. One common technique is to take 2-D images of a reference object and then estimate both intrinsic and extrinsic parameters from the 2-D images based on a camera model [1]. Using an accurate 3-D reference point, one can obtain precise camera parameters by applying 3-D reference object-based calibration algorithms [5–7]. A 2-D plane-based calibration method [8–10] is more desired because the 2-D paper on which a checker pattern is sketched is easily available, compared to a precise 3-D reference object. A camera calibration scheme that uses a simple reference object is more preferred in the real world, as it results in systems with quickly response to various camera configurations in both outdoor and indoor applications.

Self-calibration is discussed in many previous works [11–14]. It estimates camera parameters from feature points in an image scene without any reference object. However, it needs many reliable feature points and multiple image views

✉ H. K. Ardakani  
h.k.ardakani@gmail.com

A. Mousavinia  
moosavie@eetd.kntu.ac.ir

Farzad Safaei  
farzad\_safaei@uow.edu.au

<sup>1</sup> Electrical and Computer Engineering Faculty, K.N.Toosi University of Technology, Tehran, Iran

<sup>2</sup> Faculty of Engineering and Information Sciences, University of Wollongong, Wollongong, NSW, Australia

in order to extract accurately camera parameters, as a drawback.

An important type of techniques that usually use a stick consisting of at least three points to calibrate camera parameters is 1-D calibration techniques. The method using 1-D object proposed by Zhang [15] has a simpler structure than the 2-D model plane. This calibration technique uses a terminal point as a fixed reference point and rotates a 1-D object around it. For this calibration technique, at least six views of the 1-D object are required with three collinear points as reference.

Many other researchers have been inspired by Zhang's algorithm. The degenerate cases or critical motions, where the above 1-D object-based calibration fails, are studied by Hammarstedt et al. [16]. In [17], it has been shown that the rotating 1-D object is in essence equivalent to a 2-D planar calibration object, and also shown that when the object undergoes planar motion, the calibration principle still holds. Although the 1-D reference object is constrained to rotation and planar motion, Qi et al. [18] developed a camera calibration method using a 1-D object with general motion. Wang et al. [19] proposed a multi-camera calibration algorithm with a 1-D object under general motion. de Francca et al. [20] proposed a linear algorithm with the normalized image points, which significantly improved the accuracy of the calibration introduced in [15].

To avoid possible non-positive-definite values of the image of absolute conic (IAC), Wang et al. [21] minimized the norm of algebraic residuals considering to the constraint that the solution was positive definite. In [22], a calibration method using a single image and a special combination of two orthogonal 1-D objects is proposed. Each object has three collinear points, sharing a common one. The discussed calibration algorithm uses the five points to estimate the extrinsic parameters. This algorithm is very sensitive to distortion and its accuracy decreases dramatically as the distortion level increases. In [23], a more accurate algorithm based on weighted similarity invariant for camera calibration is proposed. However, it needs multiple images of rotated object to obtain the result which increases its overhead and complexity. Recently, Zhao et al. [24] have studied a camera calibration method using three non-collinear points. The algorithm needs some image of precisely rotated object to work. The final accuracy is strongly related to these rotations. The human body as a 1-D reference object also was used in order to estimate camera parameters from images of human faces [25] and walking humans [26].

It is important that the system operators have a flexible and simple calibration method to handle camera configurations and obtain camera parameters smoothly even out of laboratory. Such methods should offer a camera calibration technique using just one single simple object as reference. A well-known approach that uses a single image is the camera

calibration based on vanishing points [1]. In [27], a calibration method has been discussed that uses vanishing points to estimate intrinsic parameters from a single camera and extrinsic parameters from a pair of cameras. Next, in [28] a method for measuring intrinsic parameters using vanishing points from one image of model planes has been developed. Cipolla et al. [29] proposed a method to compute both intrinsic and extrinsic parameters from three vanishing points and two reference points from two views. The vanishing points computed from static scene structures are used for camera calibration in [30, 31]. It is possible to use orthogonal 2-D model planes as one of the means to obtain both intrinsic and extrinsic parameters by using vanishing points from a single image. That is, one can estimate the intrinsic parameters from the vanishing points detected in the image and obtain the extrinsic parameters using plane homographies [7]. All above-mentioned algorithms suffer from the lack of accuracy and high complexity.

There has been a proliferation of research on sports video analysis in the recent years [32–34]. Camera calibration also plays a vital role in sports video analysis, enabling a variety of applications, such as semantic/tactic analysis [35], 3-D ball trajectory reconstruction [36] and free viewpoint video synthesis [37]. For sports videos, instead of setting up a calibration object, a *passive* object such as the court with a known 3-D model is often exploited to compute the camera projection matrix. One classical way is based on the detection of *corresponding points*, the reference points whose coordinates are known in both the 3-D real-world and the 2-D image, such as intersection points of court lines and characteristic points on *court objects* (e.g., a net post in tennis or badminton). Then by solving a set of linear equations obtained from these corresponding points, the camera projection matrix is computed [35]. In [38], a calibration technique based on geometric analysis is proposed. This method can estimate the camera parameters from just one view of only five corresponding points. Geometric analysis is used to realize camera calibration from four coplanar corresponding points and a fifth non-coplanar one. For applications of sports video analysis, it is also easy to find four planar points (e.g., intersections of court lines) and a non-planar one (e.g., a point on a court object).

In [39] and [40], a simple rectangular reference object is used for camera calibration. The proposed technique in [39] requires a single image having two vanishing points. Using four points of the object and the two vanishing points, it provides just the intrinsic parameters of camera. In [40], a single image of a scene rectangle of an unknown aspect ratio is used for camera calibration. It uses the vanishing points to center the quad and then rectifies the image to get a rectangle. The intrinsic and extrinsic parameters of camera are then extracted with the cost of using some iterative solutions.

In this paper, a new calibration algorithm using just one image and a simple 2-D reference object with four corner points is proposed. It is absolutely practical and one can even use an A4-size paper as the reference object and is very fast as it uses just one single image. The proposed method is of high applicability in many types of scenes. For example, in an indoor scene, one will be able to set four reference points on the floor (e.g., corners of ceramic tiles) to conduct our calibration method. In an outdoor scene, one can simply choose four preselected (vs. premarked) points on the ground as reference points. For applications of sports video analysis, it is also easy to find four reference points (e.g., intersections of court lines). The proposed method is very accurate, and compared to the other ones, it has less distortion sensitivity.

This paper is organized as follows: Sect. 2 describes the calculation of camera parameters based on four corner points of a 2-D reference object. The camera calibration algorithm is discussed in Sect. 3. Section 4 provides simulations and experimental results, and finally, Sect. 5 provides conclusions.

## 2 Mathematic fundamental of propose method

Suppose points  $P'_1$  to  $P'_4$  are the corner points of the reference object in  $X', Y', Z'$  coordinate system as in Fig. 1. The corresponding points on the image plane are named as  $a_1$  to  $a_4$ . The relation between these eight points can be described by:

$$a_j = \begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} U_j \\ V_j \\ S_j \end{bmatrix} = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} X'_j \\ Y'_j \\ Z'_j \\ 1 \end{bmatrix}$$

$$x_j = \frac{U_j}{S_j} \quad y_j = \frac{V_j}{S_j}, \quad \text{for } j = 1, 2, 3, 4 \tag{1}$$

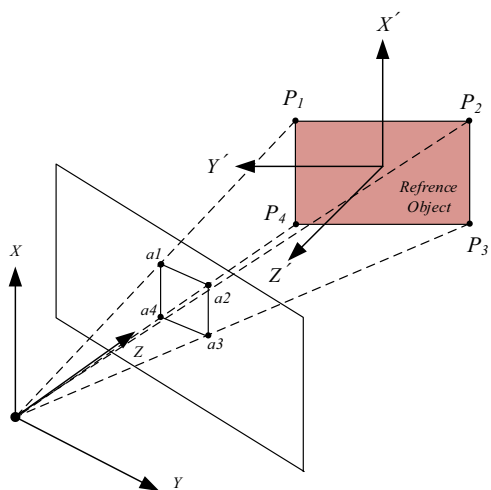


Fig. 1 Space location of camera and reference object

$$R = R_X \times R_Y \times R_Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix}$$

$$\times \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \times \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \tag{3}$$

in which  $f$  is the camera focal length,  $R$  is rotation matrix and  $T$  is camera translation vector into space coordinates (Eqs. 2, 3). We will consider that  $f$  is equal in  $x$  and  $y$  directions, principal point is projection center and there is no lens distortion to simplify equations.

It is desired to obtain extrinsic parameters of camera in space coordinates from intrinsic parameters of it. With today's technology, the image center coordinates of the camera are available and we take them as initial values.

To solve this problem, suppose that reference object is moved to the center of camera coordinates which is  $(X, Y, Z)$  by rotation matrix of  $R$  and translation matrix of  $T$ , as shown in Fig. 2. The camera coordinates of the corner points of the reference object is now as follows:

$$P'_1 = \begin{bmatrix} l_1 \\ l_2 \\ 0 \end{bmatrix} \quad P'_2 = \begin{bmatrix} -l_1 \\ l_2 \\ 0 \end{bmatrix} \quad P'_3 = \begin{bmatrix} -l_1 \\ -l_2 \\ 0 \end{bmatrix}$$

$$P'_4 = \begin{bmatrix} l_1 \\ -l_2 \\ 0 \end{bmatrix} \tag{4}$$

in which  $l_1$  and  $l_2$  are halves of reference object dimensions. Equation (5) shows the relation between  $P'_j$  and  $P_j$  points.

$$P_j = \begin{bmatrix} X_{P_j} \\ Y_{P_j} \\ Z_{P_j} \end{bmatrix} = \begin{bmatrix} R^T & -R^T T \\ 0 & 1 \end{bmatrix} \times P'_j \quad j = 1, 2, 3, 4 \tag{5}$$

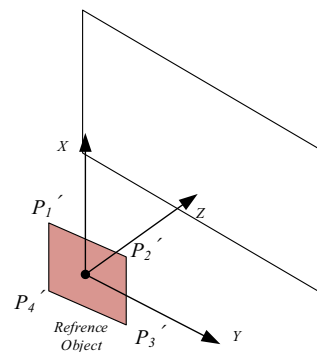


Fig. 2 Reference object transferred to camera origin coordinates

As a result, the coordinates of image points is calculated as:

$$\begin{bmatrix} U_j \\ V_j \\ S_j \end{bmatrix} = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} X_{P_j} \\ Y_{P_j} \\ Z_{P_j} \\ 1 \end{bmatrix}$$

$$a_j = \begin{bmatrix} x_{a_j} \\ y_{a_j} \end{bmatrix} = \begin{bmatrix} \frac{U_j}{S_j} \\ \frac{V_j}{S_j} \end{bmatrix} = \begin{bmatrix} \frac{f \times X_{P_j}}{Z_{P_j}} + x_0 \\ \frac{f \times Y_{P_j}}{Z_{P_j}} + y_0 \end{bmatrix},$$

$$\text{for } j = 1, 2, 3, 4 \quad (6)$$

Having the image of reference object, the  $a_j$ , and actual reference object corners, the  $P'_j$ , using (5), (6) ( $l_1, l_2$  and  $a_1$  to  $a_4$  are known) one can calculate unknown  $R, T$  and  $f$  as follows.

Defining  $B_j, a, b, c, D_{2j}, D_{1j}$  as in Eq. (7) to (2).

$$B_j = \left( \frac{x_{a_j} - x_0}{f} \right) / \left( \frac{y_{a_j} - y_0}{f} \right), \quad \text{for } j = 1 \text{ to } 4 \quad (7)$$

$$a = \cos(\theta_y) \times \cos(\theta_z) \quad (8)$$

$$b = \sin(\theta_x) \times \sin(\theta_y) \times \cos(\theta_z) \quad (9)$$

$$c = \cos(\theta_x) \times \sin(\theta_z) \quad (10)$$

$$D_{2j} = B_j \times \cot(\theta_z) - 1, \quad \text{for } j = 1 \text{ to } 4 \quad (11)$$

$$D_{1j} = -1 - B_j \times \tan(\theta_z), \quad \text{for } j = 1 \text{ to } 4 \quad (12)$$

These abstract equations are concluded:

$$(l_1 - T_X)a + (l_2 - T_Y) \left[ b + c \frac{D_{21}}{D_{11}} \right] - T_Z \left[ -b \cot(\theta_x) + c \tan(\theta_x) \frac{D_{21}}{D_{11}} \right] = 0 \quad (13)$$

$$(-l_1 - T_X)a + (l_2 - T_Y) \left[ b + c \frac{D_{22}}{D_{12}} \right] - T_Z \left[ -b \cot(\theta_x) + c \tan(\theta_x) \frac{D_{22}}{D_{12}} \right] = 0 \quad (14)$$

$$(-l_1 - T_X)a + (-l_2 - T_Y) \left[ b + c \frac{D_{23}}{D_{13}} \right] - T_Z \left[ -b \cot(\theta_x) + c \tan(\theta_x) \frac{D_{23}}{D_{13}} \right] = 0 \quad (15)$$

$$(l_1 - T_X)a + (-l_2 - T_Y) \left[ b + c \frac{D_{24}}{D_{14}} \right] - T_Z \left[ -b \cot(\theta_x) + c \tan(\theta_x) \frac{D_{24}}{D_{14}} \right] = 0 \quad (16)$$

Solving the above equations will result in:

$$Q_1 = \frac{D_{21}}{D_{11}} - \frac{D_{22}}{D_{12}} - \frac{D_{23}}{D_{13}} + \frac{D_{24}}{D_{14}} \quad (17)$$

$$Q_2 = \frac{D_{21}}{D_{11}} - \frac{D_{22}}{D_{12}} + \frac{D_{23}}{D_{13}} - \frac{D_{24}}{D_{14}} \quad (18)$$

$$K_1 = \frac{Q_1}{Q_2} \quad (19)$$

$$K_2 = \frac{4 \times l_1}{l_2 \times (K_1^2 - 1) \times Q_2} \quad (20)$$

$$K_3 = -\frac{K_2}{2} \left( \frac{D_{23}}{D_{13}} + \frac{D_{22}}{D_{12}} \right) - \frac{K_1 \times K_2}{2} \left( \frac{D_{23}}{D_{13}} - \frac{D_{22}}{D_{12}} \right) \quad (21)$$

$$K_4 = -\tan(\theta_x) \quad (22)$$

$$K_5 = l_2 \times K_1 \quad (23)$$

$$K_6 = K_3 \times [\cot(\theta_x) + \tan(\theta_x)] \quad (24)$$

$$K_7 = \frac{l_2}{2} \left[ K_2 \left( \frac{D_{21}}{D_{11}} - \frac{D_{23}}{D_{13}} \right) - K_1 \left[ 2 \times K_3 + K_2 \left( \frac{D_{21}}{D_{11}} + \frac{D_{23}}{D_{13}} \right) \right] \right] \quad (25)$$

$$T_Y = K_4 \times T_Z + K_5 \quad (26)$$

$$c = K_2 \times a \quad (27)$$

$$b = K_3 \times a \quad (28)$$

$$T_X = K_6 \times T_Z + K_7 \quad (29)$$

Define  $g_j$  variables for  $j=1-4$  as Eq. (30) and parameters  $H_1$  and  $H_2$  as in Eqs. (31) and (32).

$$g_j = x_{a_j} - x_0, \quad \text{for } j = 1 \text{ to } 4 \quad (30)$$

$$H_1 = \frac{l_1 + l_2 \times (K_2 + K_3)(1 - K_1) - K_7}{g_1} \quad (31)$$

$$H_2 = \frac{-l_1 + l_2 \times (K_2 + K_3)(1 - K_1) - K_7}{g_2} \quad (32)$$

Now substituting  $g_j, H_1$  and  $H_2$  in Eqs. (5) and (6), one can reach Eqs. (33) and (34).

$$H_1 \times f \cos(\theta_z) = (l_1 - T_X) \tan(\theta_y) - (l_2 - T_Y) \times \sin(\theta_x) - T_z \cos(\theta_x) \quad (33)$$

$$H_2 \times f \cos(\theta_z) = (-l_1 - T_X) \tan(\theta_y) - (l_2 - T_Y) \times \sin(\theta_x) - T_z \cos(\theta_x) \quad (34)$$

Subtracting (34) from (33),  $\theta_y$  can be obtained as:

$$\theta_y = \arctan\left(\frac{(H_1 - H_2) f \cos(\theta_z)}{2 \times l_1}\right) \tag{35}$$

Using (27) and (8) and (10),  $\theta_x$  is calculated as:

$$\theta_x = \arccos\left(K_2 \frac{\cos(\theta_y) \cos(\theta_z)}{\sin(\theta_z)}\right) \tag{36}$$

In the next step, using Eqs. (26), (29) and (33) will result in  $T_Z$  as:

$$T_Z = \frac{H_1 f \cos(\theta_z) + (K_7 - l_1) \tan(\theta_y) + (K_5 - l_2) \sin(\theta_x)}{-K_6 \tan(\theta_y) + K_4 \sin(\theta_x) - \cos(\theta_x)} \tag{37}$$

Till now, the rotation angle values,  $\theta_x$  and  $\theta_y$ , and translation vectors ( $T_X$   $T_Y$   $T_Z$ ) have been obtained based on the  $\theta_z$  value. Assume  $H_3$  defined as in Eqs. (38):

$$H_3 = \frac{-l_1 + l_2 \times (K_2 + K_3)(-1 - K_1) - K_7}{g_3} \tag{38}$$

Rewriting Eq. (33) for  $j = 3$ , we have:

$$H_3 \times f \cos(\theta_z) = (-l_1 - T_X) \tan(\theta_y) - (-l_2 - T_Y) \times \sin(\theta_x) - T_z \cos(\theta_x) \tag{39}$$

Considering (28), (33), (34) and (39), one can show that

$$(H_1 - H_2)(H_2 - H_3) f^2 \cos^2(\theta_z) + 4l_1 l_2 K_3 = 0 \tag{40}$$

Now the focal length  $f$  can be calculated as:

$$f = \sqrt{\frac{-4l_1 l_2 K_3}{(H_1 - H_2)(H_2 - H_3) \cos^2(\theta_z)}} \tag{41}$$

To calculate  $\theta_z$ , we can follow the same routine. Rewriting Eq. (33) for  $j = 4$  and assuming that  $H_4$  is defined as in Eqs. (42), we reach Eq. (43).

$$H_4 = \frac{l_1 + l_2 \times (K_2 + K_3)(-1 - K_1) - K_7}{g_4} \tag{42}$$

$$H_4 \times f \cos(\theta_z) = (l_1 - T_X) \tan(\theta_y) - (-l_2 - T_Y) \times \sin(\theta_x) - T_z \cos(\theta_x) \tag{43}$$

Considering Eqs. (33), (34), (41) and (43), one can show that:

$$H_1 - H_4 = H_2 - H_3 \tag{44}$$

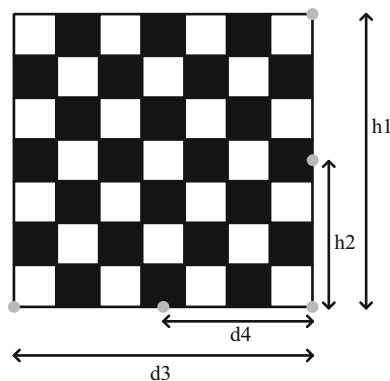


Fig. 3 Chessboard as a reference object

To simplify Eq. (44), assume that  $m_1$  to  $m_4$  are as in Eqs. (45)–(48).

$$m_1 = g_1 g_2 B_3 B_4 (B_1 - B_2) \tag{45}$$

$$m_2 = g_3 g_4 B_1 B_2 (B_3 - B_4) \tag{46}$$

$$m_3 = m_1 \left(\frac{g_4}{B_4} - \frac{g_3}{B_3}\right) + m_2 \left(\frac{g_1}{B_1} - \frac{g_2}{B_2}\right) \tag{47}$$

$$m_4 = m_1 (g_3 - g_4) + m_2 (g_2 - g_1) \tag{48}$$

Solving Eq. (48) now leads to the following equation which gives us the  $\theta_z$  value:

$$m_3 - m_4 \times \tan(\theta_z) = 0 \tag{49}$$

After finding the  $\theta_z$  value, it is now possible to find all the camera extrinsic parameters.

### 3 Proposed computing algorithm

We summarize a calibration algorithm based on the above-derived equations as follows.

1. Determining corner points of the rectangular plane from image.
2.  $B_j$  calculation using (7) and  $g_j$  calculation using (30).
3.  $\theta_z$  calculation using (49).
4.  $D_{1j}$  and  $D_{2j}$  calculation using (11) and (12).
5.  $K_1, K_2, K_3$  and  $K_7$  calculation using (19)–(21) and (25).
6.  $H_1, H_2$  and  $H_3$  calculation using (31), (32) and (38) and then  $f$  calculation using (41).
7. Determining  $\theta_y$  using (35).
8. Determining  $\theta_x$  using (36).
9.  $K_4, K_5$  and  $K_6$  calculation using (22)–(24).

**Table 1** Extrinsic parameters for simulation

$\theta_Z$	$\theta_Y$	$\theta_X$	$T_Z$	$T_Y$	$T_X$
45°	35°	25°	1.8 m	3.4 m	0.2 m

10.  $T_Z$  calculation using (37).
11.  $T_Y$  and  $T_X$  calculation using (26) and (29).

Using the previous mentioned algorithm, all of the variables are calculated. Note that if  $\theta_z$  is zero or  $\frac{\pi}{2}$ , Eqs. (11) and (12) will not result in  $D_{1j}$  and  $D_{2j}$ , so the rest of the parameters will not be obtained and the algorithm will not work. This is not important because with careful choosing 2-D reference object position in 3-D space, it can be avoided.

So far, the camera is calibrated in relation to 2-D reference object coordinates. If the space coordinates is not the same as to 2-D reference object one, the camera should be calibrated

in relation to space coordinates. To achieve this, it is sufficient to rotate and transform the space coordinates in relation to the 2-D reference object coordinates which are considered as follows:

$$\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{S2} & T_{S2} \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} R' & T' \\ 0 & 1 \end{bmatrix} \tag{50}$$

in which  $R_{S2}$  is the rotation matrix and  $T_{S2}$  is the translation vector both calculated in above and  $R'$  and  $T'$  are the rotation matrix and the translation vector of the space coordinates in relation to 2-D reference object coordinates, respectively. Note that while the position of the 2-D reference object is available,  $T'$  and  $R'$  are available. So the camera is calibrated in relation to space coordinates.

The previous mentioned algorithm could be used for a multi-camera system. Since in a multi-camera system, all the cameras will not see the object necessarily, calculation of the extrinsic parameters will be a problem [41]. Considering that

**Table 2** Simulation results for different values of distortion STD

Algorithm	[22]	Our method	[7]	[22]	Our method	[7]
	Distortion STD = 1			Distortion STD = 2		
$\Delta T_X$	0.0121	0.0193	0.0097	0.0268	0.0381	0.0209
$\Delta T_Y$	0.0942	0.0022	0.0110	0.2646	0.0043	0.0235
$\Delta T_Z$	0.1592	0.0131	0.0079	0.3816	0.0251	0.0162
$\Delta \theta_x$	4.1646	0.1207	0.1310	8.50	0.2405	0.2766
$\Delta \theta_y$	2.3324	0.1290	0.1362	7.84	0.2549	0.2827
$\Delta \theta_z$	5.5240	0.1545	0.1955	13.35	0.2999	0.4124
$\Delta f$	279	7.789	6.690	1052	15.31	14.30
Time(s)	0.177	0.060	0.463	0.210	0.052	0.456
	Distortion STD = 3			Distortion STD = 4		
$\Delta T_X$	0.055	0.0571	0.0375	0.055	0.0770	0.0375
$\Delta T_Y$	0.6678	0.0065	0.0365	0.6678	0.0084	0.0365
$\Delta T_Z$	0.7408	0.0376	0.0243	0.7408	0.0499	0.0243
$\Delta \theta_x$	13.2	0.3604	0.4870	13.2	0.4856	0.4870
$\Delta \theta_y$	17.07	0.3813	0.4830	17.07	0.5113	0.4830
$\Delta \theta_z$	25.21	0.4352	0.7225	25.21	0.5836	0.7225
$\Delta f$	1885	22.89	24.52	2422	30.11	39.80
Time(s)	0.213	0.057	0.448	0.225	0.055	0.440
	Distortion STD = 5			Distortion STD = 6		
$\Delta T_X$	0.0763	0.0975	0.0876	0.0861	0.1141	0.1255
$\Delta T_Y$	1.1411	0.0105	0.0718	1.9292	0.0122	0.0947
$\Delta T_Z$	1.617	0.0645	0.0442	2.3948	0.0758	0.0520
$\Delta \theta_x$	21.77	0.5950	1.1067	24.43	0.6891	1.5057
$\Delta \theta_y$	26.53	0.6271	1.0492	27.61	0.7187	1.4092
$\Delta \theta_z$	38.94	0.7314	1.6521	41.63	0.8936	2.2983
$\Delta f$	2509	39.15	59.66	3024	45.69	81.86
Time(s)	0.253	0.061	0.444	0.258	0.065	0.470

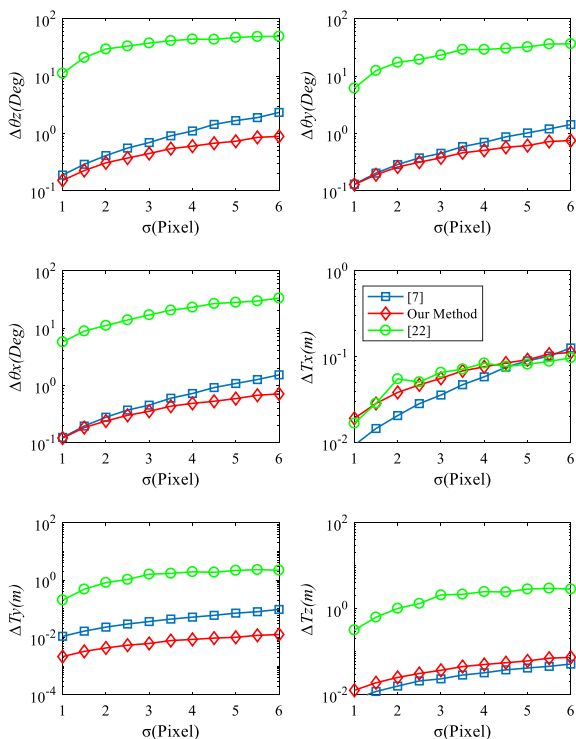


Fig. 4 Comparison of our method to the existing methods

the 2-D plane used for the calibration is seen with all cameras or several 2-D planes are used in such a way that all of the cameras see one 2-D plane at least, the proposed algorithm could be used for all of the cameras of the multi-camera system. Since every camera has its own coordination system, the extrinsic parameters of each camera should be transformed to the multi-camera coordination system. To accomplish this job, the method used in [22] could be used.

### 4 Simulation and experimental results

#### 4.1 Computer-aided simulation

The 1-D and 2-D calibration methods presented in [22] and [7] have been used in comparison with our proposed method. An ASUS N501VW laptop with 12 GB DDRIV RAM and a core i7 6700HQ CPU is used to run the algorithms. For the 2-D algorithm of Ref [7], a uniform two-dimensional array of 8 × 8 points in a rectangle of 70 × 70 cm<sup>2</sup> is considered as in Fig. 3, with  $h1 = 70$  cm,  $h2 = 35$  cm,  $d3 = 70$  cm,  $d4 = 35$  cm for the proposed object of [22]. It is assumed that camera has a focal length of 1700, unity aspect ratio, zero skew and lens distortion with the principle point at (500, 500). The image resolution is 1000 × 1000. The extrinsic parameters of camera are given in Table 1.

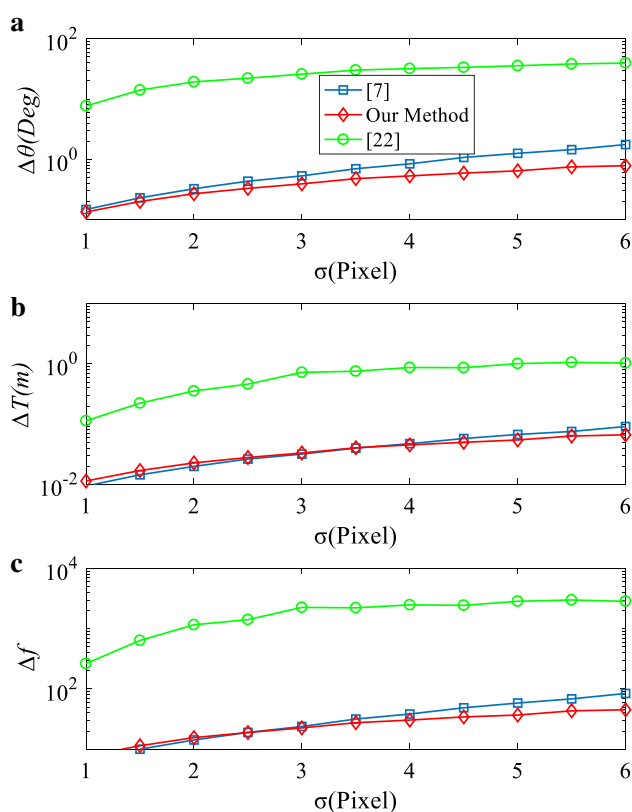


Fig. 5 Comparison of our method to the existing methods. a Average location error; b orientation error; c focal length error

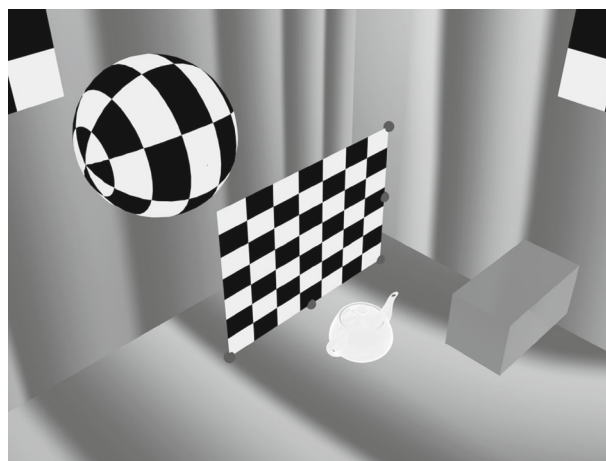
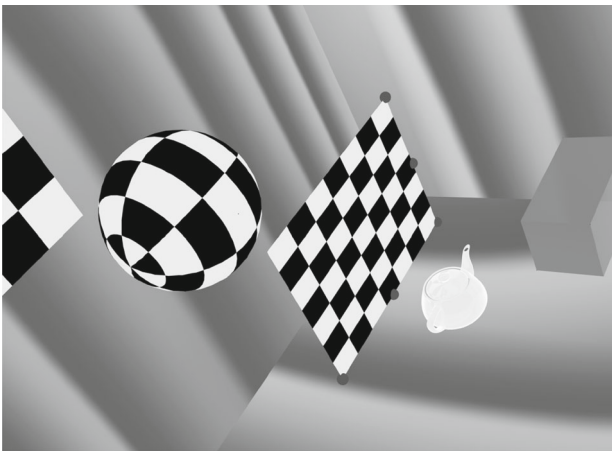


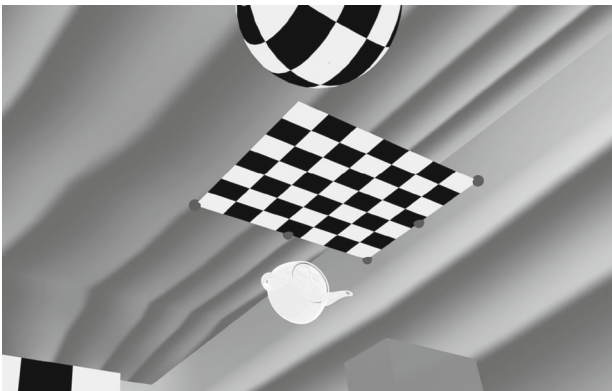
Fig. 6 Generated image by 3ds Max software for calibration

To evaluate the effect of distortion, a Gaussian noise with zero mean and different values of STD, between 1 and 6 pixels, is added to the position of 8 × 8 reference points in the image.

For each value of injected distortion, the algorithm is performed 1000 times and the absolute average error and total simulation time per each method are measured. Table 2 shows the results and it is clear that by increasing the distortion STD,



**Fig. 7** Rotated image generated by 3ds Max software for calibration algorithm comparison



**Fig. 8** Third image generated by 3ds Max software required for calibration proposed in [7]

our proposed method outperforms the rivals and in  $STD = 6$ , the maximum angle error of our algorithm is almost less than the minimum of [22] algorithm with  $STD = 1$ .

In addition, the proposed algorithm is three times faster than [22] and seven times faster than [7].

The results in Table 2 are used to draw graphs in Figs. 4 and 5 in which the vertical axes are logarithmic to present the difference. Figure 5a and b shows the average attitude

measurement error, consisting of location error and orientation error, respectively. The focal length measuring error is reported in Fig. 5c. Again one can see that the present method has least distortion sensitivity especially when the distortion increases.

## 4.2 Data calculated using Autodesk 3ds Max 2013

Here Autodesk 3ds Max 2013 is used to generate reference images as in Figs. 6, 7 and 8. In these images, the chessboard size is  $30 \times 40 \text{ cm}^2$ . For Ref. [22], parameters are  $h1 = 30 \text{ cm}$ ,  $h2 = 15 \text{ cm}$ ,  $d3 = 40 \text{ cm}$ ,  $d4 = 20 \text{ cm}$ , while Ref. [7] uses the 64 points in the chessboard as before. Calibration result comparison of three calibration methods with the image produced with Autodesk 3ds Max 2013 (Fig. 6) is given in Table 3.

Considering the previous explanations for our algorithm and its disability near  $\theta_z = 0$ , now we rotate the image in respect to z-axis which is shown in Fig. 7. Calibration result comparison is given in Table 4. The third image required for calibration method in [7] is shown in Fig. 8.

As given in Table 4, the proposed algorithm is practically more accurate than other algorithms.

## 4.3 Experimental results

In this section, the validity of the proposed method is assessed by the experimental results. For this purpose, Microsoft Camera Calibration data set is applied [42].

In this data set, the reference object which is like a chessboard has dimensions of  $17 \text{ cm} \times 17 \text{ cm}$ . Ref [7] uses all the 256 corner points in the reference object, while ref [22] uses just five points, three corner points and two on the outside edges. For this algorithm, we have  $h1 = 17 \text{ cm}$ ,  $h2 = 10.256 \text{ cm}$ ,  $d3 = 17 \text{ cm}$ ,  $d4 = 10.256 \text{ cm}$ .

Ref [7] needs at least three reference images, and the images shown in Figs. 9, 10 and 11 are used for this method. Our proposed method and Ref [22] need just one image to work. Tables 5 and 6 show the results of three calibration methods of images in Figs. 9 and 10. Ref [7] corresponding

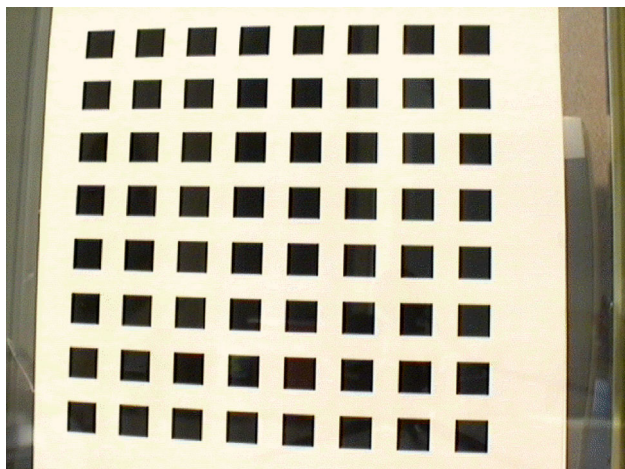
**Table 3** Calibration result comparison for image 6

	The real value	Results			Errors (%)		
		[7]	Our method	[22]	[7]	Our method	[22]
$T_X$	75	75.09	75.24	76.38	0.12	0.32	1.84
$T_Y$	-75	-74.5	-74.89	-75.05	0.79	0.15	0.07
$T_Z$	-75	-73.35	-75.05	-74.62	2.2	0.07	0.51
$\theta_X$	-45	-44.47	-44.93	-44.85	1.18	0.16	0.33
$\theta_Y$	-35	-34.69	-35.01	-35.56	0.89	0.03	1.6
$\theta_Z$	0	-0.56	0.009	-0.8	-	-	-
$f$	4830	4800	4823	4854	0.63	0.15	0.50

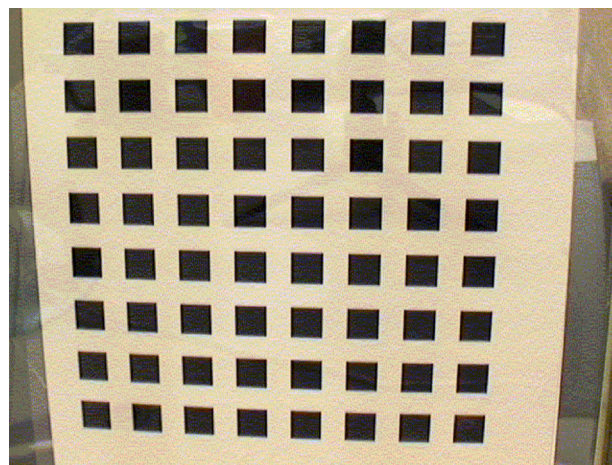


**Table 4** Calibration result comparison for image 7

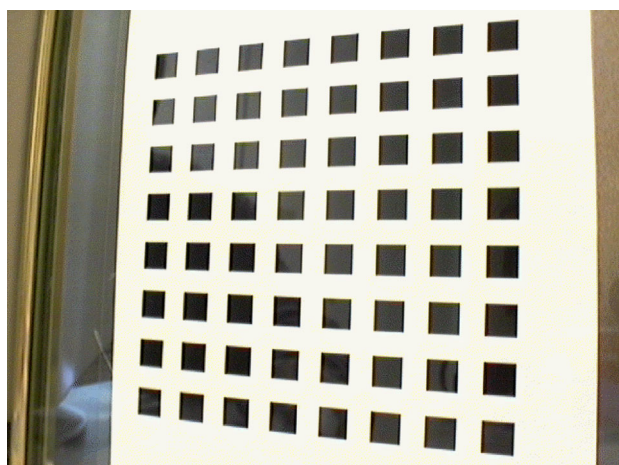
	The real value	Results			Errors (%)		
		[7]	Our method	[22]	[7]	Our method	[22]
$T_X$	75	75.78	74.90	78.5	1.04	0.13	4.67
$T_Y$	- 75	- 73.72	- 74.87	- 80.01	1.71	0.17	6.68
$T_Z$	- 75	- 74.29	- 74.90	- 82.22	0.95	0.13	9.62
$\theta_X$	- 41	- 40.3	- 40.97	- 42.46	1.71	0.07	3.56
$\theta_Y$	- 32.35	- 31.98	- 32.32	- 32.02	1.14	0.09	1.02
$\theta_Z$	28.29	28.63	28.29	27.26	1.20	0.00	3.64
$f$	4830	4800	4816	5243	0.63	0.29	8.55



**Fig. 9** Selected image from Microsoft Camera Calibration data set for calibration



**Fig. 11** Third image selected from Microsoft Camera Calibration data set required for calibration proposed in [7]



**Fig. 10** Rotated image selected from Microsoft Camera Calibration data set for calibration algorithm comparison

**Table 5** Calibration result comparison for image 9

	The real value	Results		
		[7]	Our method	[22]
$T_X$	- 3.84	- 3.87	- 3.58	- 3.40
$T_Y$	3.65	3.67	3.52	4.01
$T_Z$	12.79	12.75	12.28	12.58
$\theta_X$	0.80	1.14	0.91	1.08
$\theta_Y$	6.85	6.70	6.90	7.46
$\theta_Z$	- 5.95	- 5.99	- 5.94	- 3.98
$f$	832.5	830.1	784.3	760.8

camera z-axis, our algorithm works better and the results are similar to those of Ref [7] as given in Table 6 which is related to Fig. 10. We must consider that the image used in data set has a lot of lens distortion which reduces the accuracy of our method.

data in Tables 5 and 6 have similar intrinsic parameters, while there are extrinsic data in Figs. 9 and 10.

Looking carefully at Tables 5 and 6, we can say that our proposed method accuracy is somewhere between Refs [7] and [22]. When the reference object normal is not along the

### 5 Conclusion

In these papers, a mathematic calibration method to find all extrinsic and intrinsic parameters of a camera using just

**Table 6** Calibration result comparison for image 10

	The real value	Results		
		[7]	Our method	[22]
$T_X$	- 2.94	- 2.98	- 2.51	- 2.40
$T_Y$	3.78	3.79	3.69	4.17
$T_Z$	14.25	14.19	14.07	13.95
$\theta_x$	- 0.51	0.79	- 0.28	2.99
$\theta_y$	23.76	23.62	23.79	22.96
$\theta_z$	- 6.33	- 6.15	- 6.35	- 10.91
$f$	832.5	830.1	865.2	821.3

one 2-D reference object is proposed. Then the accuracy of the method during different distortions is evaluated. Using images obtained with 3ds Max software, the algorithm is simulated. Finally the method is applied to Microsoft Camera Calibration data set for obtaining experimental results. For comparison, works in [7, 22] are considered.

Considering the results, the proposed method has the least distortion sensitivity and computational load, while its accuracy is slightly less than the 2-D calibration method and much better than 1-D ones. In addition, the used reference object is the most simple and available among other rivals and so the proposed method has suitable for applications of sports video analysis.

## Appendix

### How to get a relationship from (13)–(16) to (17)–(35)

Summing Eqs. (13) and (14), we have:

$$\begin{aligned}
 & -2T_X a + (l_2 - T_Y) \left[ 2b + c \left( \frac{D_{21}}{D_{11}} + \frac{D_{22}}{D_{12}} \right) \right] \\
 & - T_Z \left[ -2b \cot(\theta_x) + c \tan(\theta_x) \left( \frac{D_{21}}{D_{11}} + \frac{D_{22}}{D_{12}} \right) \right] = 0 \quad (51)
 \end{aligned}$$

Summing Eqs. (13) and (15), we have:

$$\begin{aligned}
 & -2T_X a + l_2 c \left( \frac{D_{21}}{D_{11}} - \frac{D_{23}}{D_{13}} \right) - T_Y \left[ 2b + c \left( \frac{D_{21}}{D_{11}} + \frac{D_{23}}{D_{13}} \right) \right] \\
 & - T_Z \left[ -2b \cot(\theta_x) + c \tan(\theta_x) \left( \frac{D_{21}}{D_{11}} + \frac{D_{23}}{D_{13}} \right) \right] = 0 \quad (52)
 \end{aligned}$$

Summing Eqs. (13) and (16), we have:

$$\begin{aligned}
 & 2(l_1 - T_X) a + l_2 c \left( \frac{D_{21}}{D_{11}} - \frac{D_{24}}{D_{14}} \right) \\
 & - T_Y \left[ 2b + c \left( \frac{D_{21}}{D_{11}} + \frac{D_{24}}{D_{14}} \right) \right]
 \end{aligned}$$

$$- T_Z \left[ -2b \cot(\theta_x) + c \tan(\theta_x) \left( \frac{D_{21}}{D_{11}} + \frac{D_{24}}{D_{14}} \right) \right] = 0 \quad (53)$$

Summing Eqs. (14) and (15), we have:

$$\begin{aligned}
 & 2(-l_1 - T_X) a + l_2 c \left( \frac{D_{22}}{D_{12}} - \frac{D_{23}}{D_{13}} \right) \\
 & - T_Y \left[ 2b + c \left( \frac{D_{22}}{D_{12}} + \frac{D_{23}}{D_{13}} \right) \right] \\
 & - T_Z \left[ -2b \cot(\theta_x) + c \tan(\theta_x) \left( \frac{D_{22}}{D_{12}} + \frac{D_{23}}{D_{13}} \right) \right] = 0 \quad (54)
 \end{aligned}$$

Summing Eqs. (14) and (16), we have:

$$\begin{aligned}
 & -2T_X a + l_2 c \left( \frac{D_{22}}{D_{12}} - \frac{D_{24}}{D_{14}} \right) - T_Y \left[ 2b + c \left( \frac{D_{22}}{D_{12}} + \frac{D_{24}}{D_{14}} \right) \right] \\
 & - T_Z \left[ -2b \cot(\theta_x) + c \tan(\theta_x) \left( \frac{D_{22}}{D_{12}} + \frac{D_{24}}{D_{14}} \right) \right] = 0 \quad (55)
 \end{aligned}$$

Now subtracting Eq. (52) from (55) and keeping in mind that  $Q_1, Q_2, K_1, K_2, K_4, K_5$  are defined as Eqs. (17–20) and (22–23), we reach Eq. (26). In the same way, Eq. (27) is obtained from subtracting Eq. (53) from (54).

Substituting Eqs. (22), (23), (26) and (27) in Eqs. (51) and (52), one can reach Eqs. (56) and (57), respectively.

$$\begin{aligned}
 & -2T_X a + l_2(1 - K_1) \left[ 2b + K_2 a \left( \frac{D_{21}}{D_{11}} + \frac{D_{22}}{D_{12}} \right) \right] \\
 & + 2bT_Z [\cot(\theta_x) + \tan(\theta_x)] = 0 \quad (56)
 \end{aligned}$$

$$\begin{aligned}
 & -2T_X a + l_2 K_2 a \left( \frac{D_{22}}{D_{12}} - \frac{D_{24}}{D_{14}} \right) \\
 & - l_2 K_1 \left[ 2b + K_2 a \left( \frac{D_{22}}{D_{12}} + \frac{D_{24}}{D_{14}} \right) \right] \\
 & + 2bT_Z [\cot(\theta_x) + \tan(\theta_x)] = 0 \quad (57)
 \end{aligned}$$

Again subtracting Eq. (56) from (57) and keeping in mind that  $K_3$  is defined as Eq. (21), we reach Eq. (28). Finally with the assumption that  $K_6, K_7$  as defined in Eqs. (24, 25), substituting Eq. (28) in Eq. (57) one can reach Eq. (29).

## References

- Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, 2nd edn. Cambridge Univ. Press, Cambridge (2003)
- Shi, J., Sun, Z., Bai, S.: 3D reconstruction framework via combining one 3D scanner and multiple stereo trackers. *Vis. Comput.* **34**(3), 377–389 (2018)
- Lu, F., Zhou, B., Zhang, Y., et al.: Real-time 3D scene reconstruction with dynamically moving object using a single depth camera. *Vis. Comput.* **34**(6–8), 753–763 (2018)
- El Hazzat, S., Merras, M., El Akkad, N., et al.: 3D reconstruction system based on incremental structure from motion using a camera with varying parameters. *Vis. Comput.* **34**(10), 1443–1460 (2017)

5. Abdel-Aziz, Y.I., Karara, H.M., Hauck, M.: Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogramm. Eng. Remote Sens.* **81**(2), 103–107 (2015)
6. Xu, G., et al.: Three degrees of freedom global calibration method for measurement systems with binocular vision. *J. Opt. Soc. Korea* **20**(1), 107–117 (2016)
7. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(11), 1330–1334 (2000)
8. Frosio, I., Turrini, C., Alzati, A.: Camera re-calibration after zooming based on sets of conics. *Vis. Comput.* **32**(5), 663–674 (2016)
9. Liu, M., et al.: Generic precise augmented reality guiding system and its calibration method based on 3d virtual model. *Opt. Express* **24**(11), 12026–12042 (2016)
10. Xu, G., Zhang, X., Su, J., Li, X., Zheng, A.: Solution approach of a laser plane based on Plücker matrices of the projective lines on a flexible 2D target. *Appl. Optics* **55**(10), 2653–2656 (2016)
11. Maybank, S.J., Faugeras, O.D.: A theory of self-calibration of a moving camera. *Int. J. Comput. Vis.* **8**(2), 123–151 (1992)
12. Triggs, B.: Auto calibration and the absolute quadric. In: *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, pp. 609–614 (1997)
13. Hemayed, E.E.: A survey of camera self-calibration. In: *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, pp. 351–357 (2003)
14. Ackermann, H., Kanatani, K.: Robust and efficient 3-D reconstruction by self-calibration. In: *Proc. IAPR Conf. Mach. Vis. Applications*, pp. 178–181 (2007)
15. Zhang, Z.: Camera calibration with one-dimensional objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(7), 892–899 (2004)
16. Hammarstedt, P., Sturm, P., Heyden, A.: Degenerate cases and closed-form solutions for camera calibration with one-dimensional objects. In: *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, pp. 317–324 (2005)
17. Wu, F.C., Hu, Z.Y., Zhu, H.J.: Camera calibration with moving one-dimensional objects. *Pattern Recognit.* **38**(5), 755–765 (2005)
18. Qi, F., Li, Q., Luo, Y., Hu, D.: Constraints on general motions for camera calibration with one-dimensional objects. *Pattern Recognit.* **40**(6), 1785–1792 (2007)
19. Wang, L., Wu, F.C., Hu, Z.Y.: Multi-camera calibration with one-dimensional object under general motions. In: *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1–7 (2007)
20. de Francca, J.A., Stemmer, M.R., Francca, M.B.D.M., Alves, E.G.: Revisiting Zhang's 1D calibration algorithm. *Pattern Recognit.* **43**(3), 1180–1187 (2010)
21. Wang, L., Duan, F., Liang, C.: A global optimal algorithm for camera calibration with one-dimensional objects. In: *Proc. 14th Int. Conf. Human-Comput. Interact.*, pp. 660–669 (2011)
22. Miyagawa, I., Arai, H., Koike, H.: Simple camera calibration from a single image using five points on two orthogonal 1-D objects. *IEEE Trans. Image Process.* **19**(6), 1528–1538 (2010)
23. Shi, K., Dong, Q., Wu, F.: Weighted similarity-invariant linear algorithm for camera calibration with rotating 1-D objects. *IEEE Trans. Image Process.* **21**(8), 3806–3812 (2012)
24. Zhao, Z., Liu, Y., Zhang, Z.: Camera calibration with three non-collinear points under special motions. *IEEE Trans. Image Process.* **17**(12), 2393–2402 (2008)
25. Cao, X., Foroosh, H.: Camera calibration using symmetric objects. *IEEE Trans. Image Process.* **15**(11), 3614–3619 (2006)
26. Lv, F., Zhao, T., Nevatia, R.: Camera calibration from video of a walking human. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(9), 1513–1518 (2006)
27. Caprile, B., Torre, V.: Using vanishing points for camera calibration. *Int. J. Comput. Vis.* **4**(2), 127–139 (1990)
28. Beardsley, P., Murray, D.: Camera calibration using vanishing points. In: *Proc. Brit. Mach. Vis. Conf.*, pp. 416–425 (1992)
29. Cipolla, R., Drummond, T., Robertson, D.: Camera calibration from vanishing points in images of architectural scenes. In: *Proc. Brit. Mach. Vis. Conf.*, vol. 2, pp. 382–391 (1999)
30. Grammatikopoulos, L., Karras, G., Petsa, E., Kalisperakis, I.: An automatic approach for camera calibration from vanishing points. *ISPRS J. Photogramm. Remote Sens.* **62**(1), 64–76 (2007)
31. Wang, G., Tsui, H.T., Hu, Z., Wu, F.: Camera calibration and 3-D reconstruction from a single view based on scene constraints. *Image Vis. Comput.* **23**(3), 311–323 (2005)
32. Babaguchi, N., Kawai, Y., Kitahashi, T.: Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE Trans. Multimed.* **4**(1), 68–75 (2002)
33. Xu, C., Wang, J., Lu, H., Zhang, Y.: A novel framework for semantic annotation and personalized retrieval of sports video. *IEEE Trans. Multimed.* **10**(3), 325–329 (2008)
34. Zhu, G., Xu, C., Huang, Q., Rui, Y., Jiang, S., Gao, W., Yao, H.: Event tactic analysis based on broadcast sports video. *IEEE Trans. Multimed.* **11**(1), 49–67 (2009)
35. Hu, M.-C., Chang, M.-H., Wu, J.-L., Chi, L.: Robust camera calibration and player tracking in broadcast basketball video. *IEEE Trans. Multimed.* **13**(2), 266–279 (2011)
36. Chen, H.-T., Tsai, W.-J., Lee, S.-Y., Yu, J.-Y.: Ball tracking and 3D trajectory approximation with applications to tactics analysis from single-camera volleyball sequences. *Multimed. Tools Appl.* **60**(3), 641–667 (2012)
37. Inamoto, N., Saito, H.: Free viewpoint video synthesis and presentation of sporting events for mixed reality entertainment. In: *Proc. ACM SIGCHI Int. Conf. Adv. Comput. Entertainment Technol.*, pp. 42–50 (2004)
38. Chen, H.-T.: Geometry-based camera calibration using five-point correspondences from a single image. *IEEE Trans. Circuits Syst. Video Technol.* **27**(12), 2555–2566 (2017)
39. Avinash, N., Murali, S.: Perspective geometry based single image camera calibration. *J. Math. Imaging Vis.* **30**(3), 221–230 (2007)
40. Lee, J.-H.: Camera calibration from a single image based on coupled line cameras and rectangle constraint. In: *Int. Conf. on Pattern Recognit.* (2012)
41. Bajramovic, F., Denzler, J.: Global uncertainty-based selection of relative poses for multi camera calibration. In: *Proc. Brit. Mach. Vis. Conf.*, vol. 2, pp. 382–391 (2008)
42. Kang, S.B., Zhang, Z.: A flexible new technique for camera calibration, Microsoft Camera Calibration data set. <https://www.microsoft.com/en-us/research/project/a-flexible-new-technique-for-camera-calibration-2/>. Accessed 1999

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**H. K. Ardakani** was born in Yazd, Iran, in 1988. He received the B.S. degree in electrical engineering from Yazd University, Iran, in 2009. From 2009 to 2011, he has been the M.S. student of the Department of Electrical Engineering, K. N. Toosi University of Technology and he is currently working toward the Ph.D. degree in the Department of Electrical Engineering, K. N. Toosi University of Technology. His research interests are machine vision, multi-view and stereo image

processing, and digital signal processing.



**Farzad Safaei** graduated from the University of Western Australia with the degree of Bachelor of Engineering (electronics) and obtained his Ph.D. in telecommunications engineering from Monash University, Australia. Currently, he is the Professor of Telecommunications Engineering at the University of Wollongong. Farzad's research interests include multimedia signal processing and communications technology.



**A. Mousavinia** received his B.S. degree in electronic engineering with honors from Ferdowsi University of Mashad, Iran, in 1992 and his M.S. degree in electrical engineering from Amir Kabir University (Tehran) in 1995. He continued his education and received his Ph.D. in electronics from Iran University of Science and Technology in 2001. He has been a Member of Academic staff with Electrical and Computer Engineering Departments of K. N. Toosi University of

Technology, Tehran, Iran, since 2005. His research interests include multi-view and stereo image processing, and digital signal processing.