Multi-Camera Calibration Using a Globe

Rui Shen, Irene Cheng, Anup Basu

Multi-Camera Calibration Using a Globe

Rui Shen, Irene Cheng, and Anup Basu

Department of Computing Science, University of Alberta, Canada
{rshen,lin,anup}@cs.ualberta.ca

Abstract. The need for calibration of multiple cameras working together in a network, or for the acquisition of free viewpoint video for 3D TV, is becoming increasingly important in recent years. In this paper we present a novel approach for calibrating multiple cameras using an ordinary globe that is usually available in every household. This method makes it possible to reduce multi-camera calibration to a level that is attainable by non-technical users. Our technique requires only one view of the globe for each camera, unlike earlier approaches that need multiple views of a calibration object. Previous methods that use spherical calibration objects can only calibrate the intrinsic parameters of cameras, but our technique is capable of calibrating both the intrinsic and extrinsic parameters simultaneously in a multi-camera setup. Experimental results verifying the accuracy of our algorithm are also presented.

1 Introduction

Camera calibration is an important step in computer vision in order to recover the geometry of a given camera (or cameras) [1]. Based on the dimension of the calibration object, most calibration techniques can be classified into four categories: 3D reference object-based calibration [2], which employs a calibration object that has precise 3D geometric features; 2D plane-based calibration [3], which uses a specially-designed planar pattern; 1D object-based calibration [4], which employs a 1D stick with three or more marked points; and self-calibration [5], which does not use a specific calibration object but determines a camera's intrinsic parameters by moving the camera in a static scene. Classical calibration techniques (*e.g.*, [6]) require predefined patterns and static cameras, and often involve solving complicated equations. Active camera calibration [7] [8] has been receiving more and more attention with the increasing use of active systems in various applications, such as surveillance and videoconference. Taking advantage of a camera's controllable movements, the camera can be automatically calibrated. In a multiple-camera setup, the relative positions and orientations of the cameras (*i.e.*, extrinsic parameters) also need to be estimated. In such a scenario, self-calibration is not sufficient to retrieve both the intrinsic and extrinsic parameters simultaneously. Therefore, a calibration object is always employed [9] [10]. We propose a multi-camera calibration algorithm using a globe as the calibration object. The widespread availability of globes makes our algorithm easy to apply in everyday environments. Several

algorithms [2] [11] [12] using spherical objects as the calibration object are proposed by previous researchers. However, these algorithms use the contour of the spheres for calibration, and they can only calibrate for the intrinsic parameters in a multi-camera set up. Our algorithm utilizes the grids on the globe for calibration, and only one view of the globe from each camera is sufficient for calibrating both the intrinsic and extrinsic parameters in a multi-camera setup.

The remainder of this paper is organized as follows: Section 2 covers background theory related to multi-camera calibration. The proposed algorithm is discussed in Section 3. Experimental results are outlined in Section 4, before the work in concluded in Section 5.

2 Preliminaries

The pinhole camera model is employed. $\boldsymbol{p} = (x, y)^T$ denotes the 2D projection of the 3D point $\boldsymbol{P} = (X, Y, Z)^T$ on the image plane. Using homogeneous coordinates, $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{P}}$ have the following relationship:

$$\lambda \tilde{\boldsymbol{p}} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\tilde{\boldsymbol{P}} \tag{1}$$

where $\boldsymbol{K} = \begin{pmatrix} \alpha & s & x_0 \\ 0 & \beta & y_0 \\ 0 & 0 & 1 \end{pmatrix}$ is the *camera calibration matrix*; λ is the projection depth; and $\boldsymbol{R}$ and $\boldsymbol{t}$ are the 3x3 rotation matrix and the 3x1 translation vector respectively; α and β are the focal lengths in pixels in the x- and y-direction respectively; and s is the skew. If the sensor element is rectangular, s becomes 0. $\gamma = \frac{\alpha}{\beta}$ is called the aspect ratio. If the sensor element is quadratic, γ becomes 1, *i.e.*, $\alpha = \beta$. (x_0, y_0) is the coordinates of the principal point on the image plane. The units of α, β and (x_0, y_0) are all in pixels. The five parameters in $\boldsymbol{K}$ are called the camera intrinsic parameters; the three Euler angles in $\boldsymbol{R}$ and the three offsets in $\boldsymbol{t}$ are called the camera extrinsic parameters. The goal of our calibration algorithm is to determine the five intrinsic and the six extrinsic parameters; no radial distortion is considered.

Camera calibration using 1D object was first proposed by Zhang [13]. A stick with at least three marked points is used as the calibration object. The locations of the points on the stick and the length of the stick are known. When the stick moves with one point fixed, the camera's intrinsic parameters can be computed using six or more observations. Suppose there is a stick $\boldsymbol{AB}$ ($\boldsymbol{A}$ is the fixed point with depth z_A) with length L (*i.e.*, $\| \boldsymbol{B} - \boldsymbol{A} \| = L$) and a third point $\boldsymbol{C}$ between $\boldsymbol{A}$ and $\boldsymbol{B}$ with $\boldsymbol{C} = \lambda_A \boldsymbol{A} + \lambda_B \boldsymbol{B}$, the following equation can be obtained:

$$z_A^2 \boldsymbol{h}^T \boldsymbol{K}^{-T} \boldsymbol{K}^{-1} \boldsymbol{h} = L^2 \tag{2}$$

with

$$\boldsymbol{h} = \tilde{\boldsymbol{a}} + \frac{\lambda_A (\tilde{\boldsymbol{a}} \times \tilde{\boldsymbol{c}}) \cdot (\tilde{\boldsymbol{b}} \times \tilde{\boldsymbol{c}})}{\lambda_B (\tilde{\boldsymbol{b}} \times \tilde{\boldsymbol{c}}) \cdot (\tilde{\boldsymbol{b}} \times \tilde{\boldsymbol{c}})} \tilde{\boldsymbol{b}} \tag{3}$$

where $\boldsymbol{K}$ is the camera calibration matrix; $\tilde{\boldsymbol{a}}$, $\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{c}}$ are the images of points $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ respectively. From Equation (3), $\boldsymbol{h}$ for the set of points under every observation can be computed. Let:

$$\boldsymbol{B} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1} = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} \tag{4}$$

with $\boldsymbol{b} = (B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33})^T$, and let $\boldsymbol{h} = (h_1, h_2, h_3)^T$, and $\boldsymbol{x} = z_A^2\boldsymbol{b}$, then Equation (2) can be rewritten as:

$$\boldsymbol{v}^T\boldsymbol{x} = L^2 \tag{5}$$

with

$$\boldsymbol{v} = (h_1^2, 2h_1h_2, h_2^2, 2h_1h_3, 2h_2h_3, h_3^2)^T. \tag{6}$$

After the $\boldsymbol{v}$'s are computed, $\boldsymbol{x}$ can be obtained by solving the equation:

$$\boldsymbol{V}\boldsymbol{x} = L^2\boldsymbol{1} \tag{7}$$

Now that $\boldsymbol{x}$ is known, the intrinsic parameters can be computed easily from Equation (4). The critical motions of the 1D object are identified by Hammarstedt *et al.* [4]. Let $\boldsymbol{v}_j$ be the vanishing point in the jth image, then the motion is critical when the vanishing points of the stick lie on a conic ω' so that:

$$\forall j \; \boldsymbol{v}_j^T\omega'\boldsymbol{v}_j = 0 \tag{8}$$

In 3D space, this means that the trajectory of the stick lies on a quadric surface.

Wang *et al.* [14] applied 1D object calibration to a multi-camera setup, and proved that this object can take more general motions instead of constrained motions. However, one constraint still applies to the motions, *i.e.*, the infinite points of the stick should not lie on a conic ω''. Let $\boldsymbol{V}_j$ be the infinite point of the stick $\boldsymbol{A}_j\boldsymbol{B}_j$ in the jth image, *i.e.*, $\tilde{\boldsymbol{V}}_j = \tilde{\boldsymbol{B}}_j - \tilde{\boldsymbol{A}}_j$, then the constraint can be expressed as the following equation NOT being satisfied:

$$\forall j \; \boldsymbol{V}_j^T\omega''\boldsymbol{V}_j = 0 \tag{9}$$

Proposition 1: The motion constraint expressed in Equation (8) is equivalent to that in Equation (9). (See Appendix for proof.)

From Proposition 1, we can see that the only motion constraint that Wang *et al.* [14] have relaxed is that no point on the stick needs to be fixed during the motion. As stated in [4], the critical motions do not depend on the actual position of the stick's fixed point, but only depend on the stick's orientation. Therefore, the direction vectors of the stick should not lie on a quadric surface during the motion.

3 Calibration Algorithm

We propose a technique to calibrate both single and multiple cameras from a single image of a globe (or sphere with evenly distributed surface points).

The intrinsic parameters of a single camera can be calibrated from the captured intersections of the constant latitude and longitude lines of a globe. Although the calibration object is now a 3D sphere, the actual computation of the cameras' parameters can mostly follow the way in which the 1D stick [13] is used.

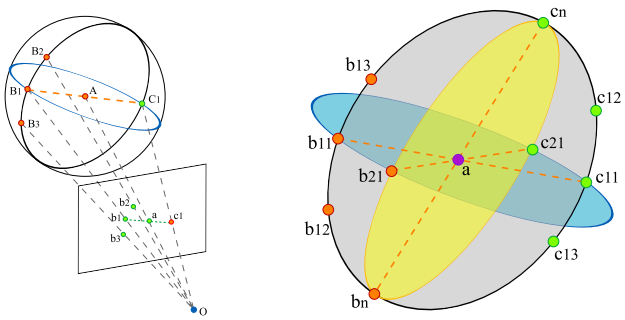

Fig. 1. (Left): The spherical calibration object; (right): image of a globe.

A globe (sphere) with marked longitude-latitude intersections is shown in Figure 1(right). $\boldsymbol{A}$ is the center of the globe, which is also the fixed point. $\boldsymbol{B_i}$'s are the intersections distributed evenly on one great circle of the globe (15° or 30° apart from each other). Every diameter $\boldsymbol{B_iAC_i}$ of a great circle is equivalent to the 1D stick at a different orientation with $\lambda_A = 2.0$ and $\lambda_B = -1.0$. Note that now L equals the radius of the globe. In the case of a globe, at least three great circles (*e.g.*, two meridians and the equator) should be used in order to avoid the degenerate cases and calibrate the cameras using only one view.

Figure 1(left) depicts the image of the globe; $\boldsymbol{b_i}$, $\boldsymbol{c_i}$, and $\boldsymbol{a}$ are the images of their corresponding spatial points respectively. Only $\boldsymbol{b_i}$'s can be obtained directly from the image, $\boldsymbol{c_i}$'s and $\boldsymbol{a}$ are not visible on the image. However, in order to calibrate the camera, the positions of all these image points need to be known. Normally, the 2D image of a 3D circle is an ellipse (if not degenerated to a line segment), which is a special case of a conic. However, in general, the center of the sphere does not project to the center of the ellipse. Using five points, the conic can be determined up to a scale factor. Let $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ denote the conics determined by the images of the equator circle and two meridians respectively. $\mathcal{L}_1 = \mathcal{C}_1 \bigcap \mathcal{C}_2$ is the line determined by the two intersection points of $\mathcal{C}_1$ and $\mathcal{C}_2$, and $\mathcal{L}_2 = \mathcal{C}_1 \bigcap \mathcal{C}_3$ is the line determined by the two intersection points of $\mathcal{C}_1$ and $\mathcal{C}_3$. $\boldsymbol{a}$ is determined by the intersection of $\mathcal{L}_1$ and $\mathcal{L}_2$, *i.e.*, $\boldsymbol{a} = \mathcal{L}_1 \bigcap \mathcal{L}_2$. Once $\boldsymbol{a}$ is calculated, one can compute each $\boldsymbol{c_i}$ by calculating the line-conic in-

tersections. Let $\boldsymbol{b}_{ij}$ denote the jth image point on the conic $\mathcal{C}_i$, and let $\boldsymbol{b}_{ij} + \lambda \boldsymbol{a}$ denote a point on the line defined by $\boldsymbol{b}_{ij}$ and $\boldsymbol{a}$. Then, the other intersection of the line with $\mathcal{C}_i$ is $\boldsymbol{c}_{ij}$ with $\lambda = -(\boldsymbol{b}_{ij}^T \mathcal{C}_i \boldsymbol{b}_{ij})/2(\boldsymbol{b}_{ij}^T \mathcal{C}_i \boldsymbol{a})$.

From here, we can follow the method proposed by Zhang [13] to recover the intrinsic parameters. However, part of the final calculation of the intrinsic parameters from $\boldsymbol{x}$ has some inaccuracies in [13]. Here, we give more accurate equations:

$$y_0 = (x_2 x_4 - x_1 x_5)/(x_1 x_3 - x_2^2) \tag{10}$$

$$z_A = \sqrt{x_6 - [x_4^2 + y_0(x_2 x_4 - x_1 x_5)]/x_1} \tag{11}$$

$$\alpha = \sqrt{z_A^2/x_1} \tag{12}$$

$$\beta = \sqrt{z_A^2 x_1/(x_1 x_3 - x_2^2)} \tag{13}$$

$$s = -x_2 \alpha^2 \beta / z_A^2 = -x_2 \beta / x_1 \tag{14}$$

$$x_0 = s y_0/\beta - x_4 \alpha^2 / z_A^2 = -(x_2 y_0 + x_4)/x_1 \tag{15}$$

At least three great circles (one equator and two meridians) are needed to avoid degenerate cases, and at least five points of each great circle need to be visible to determine the conic. Therefore, at least fifteen points are needed. However, two points can be used twice, which are the intersections of the equator and the meridians. Hence, thirteen visible points are the minimal requirement for calibration.

Proposition 2: For a set of image points of a camera, if there exists an L that can be used to derive a $\boldsymbol{K}$, then for any $L > 0$ the algorithm produces the same $\boldsymbol{K}$. (See Appendix for proof.)

Proposition 2 means that using a set of images, the radius L of the globe does not need to be known in order to get the correct calibration result. In addition, even if the value of the radius is assigned arbitrarily, the results are consistent. However, different assignments of L do affect the reconstruction, because different projection depths (z_A) are obtained.

3.1 Multi-Camera Calibration

In the previous section, the calibration of a single camera using a globe is discussed. Multiple cameras can also be calibrated simultaneously by capturing the positions of intersections of the constant latitude and longitude lines of a globe. Normally, one camera is selected as the reference camera, the coordinate system of which is assumed to coincide with the world coordinate system. The positions and orientations of other cameras are calculated relative to this reference camera.

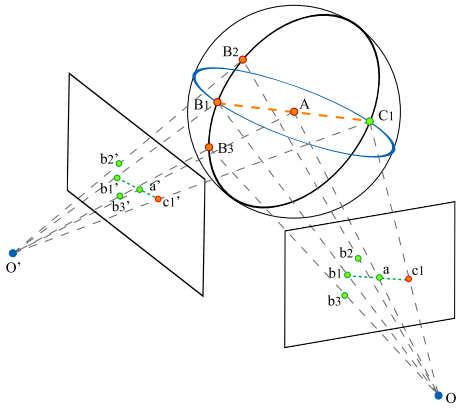

Fig. 2. The spherical calibration object in a multi-camera setup.

The spherical calibration object is shown in a two-camera case in Figure 2. Once the position and orientation of one camera relative to its adjacent camera are calculated, by combining the transformations from camera to camera, the position and orientation of one camera relative to any other camera can be obtained. Therefore, in this section we only discuss the two-camera case.

Suppose there are two cameras, Camera_0 with projection matrix $\boldsymbol{K}_0[\boldsymbol{I} \mid 0]$, and Camera_1 with projection matrix $\boldsymbol{K}_1\boldsymbol{R}_1[\boldsymbol{I} \mid -\boldsymbol{O}_1] = \boldsymbol{K}_1[\boldsymbol{R}_1 \mid \boldsymbol{t}_1]$, where $\boldsymbol{O}_1$ is the coordinate of Camera_1's center in the world (Camera_0's) coordinate system and $\boldsymbol{t}_1 = -\boldsymbol{R}_1\boldsymbol{O}_1$. $\boldsymbol{K}_0$ and $\boldsymbol{K}_1$ can be calibrated using the method proposed in the previous section. Hence, the remaining problem is to calculate $\boldsymbol{R}_1$ and $\boldsymbol{O}_1$. The projection expressed in Equation (1) can also be written as:

$$\lambda\tilde{\boldsymbol{p}} = \boldsymbol{K}[\boldsymbol{I} \mid 0]\tilde{\boldsymbol{P}}_{cam} \tag{16}$$

where

$$\tilde{\boldsymbol{P}}_{cam} = \begin{pmatrix} \boldsymbol{R} & -\boldsymbol{R}\boldsymbol{O} \\ 0 & 1 \end{pmatrix} \tilde{\boldsymbol{P}} \tag{17}$$

is the position of point $\boldsymbol{P}$ in the camera's own coordinate system using homogeneous representation. When non-homogeneous representation is used,

$$\boldsymbol{P}_{\boldsymbol{cam}} = \boldsymbol{R}(\boldsymbol{P} - \boldsymbol{O}). \tag{18}$$

Since z_A is calculated together with K, z_{B_i}'s can be computed from the following equation as in [13]:

$$z_{B_i} = -z_A \frac{\lambda_A(\tilde{\boldsymbol{a}} \times \tilde{\boldsymbol{c}_i}) \cdot (\tilde{\boldsymbol{b}_i} \times \tilde{\boldsymbol{c}_i})}{\lambda_B(\tilde{\boldsymbol{b}_i} \times \tilde{\boldsymbol{c}_i}) \cdot (\tilde{\boldsymbol{b}_i} \times \tilde{\boldsymbol{c}_i})}. \tag{19}$$

Then, the spatial points can be reconstructed in each camera's own coordinate system using a variation of Equation (16), which is

$$\tilde{\boldsymbol{P}}_{cam} = \begin{pmatrix} \lambda\boldsymbol{K}^{-1}\tilde{\boldsymbol{p}} \\ 1 \end{pmatrix}. \tag{20}$$

Since $\tilde{\boldsymbol{B}}_{i,cam0} = \boldsymbol{I}\tilde{\boldsymbol{B}}_i = \tilde{\boldsymbol{B}}_i$,
and $\tilde{\boldsymbol{B}}_{i,cam1} = \begin{pmatrix} \boldsymbol{R}_1 & -\boldsymbol{R}_1\boldsymbol{O}_1 \\ 0 & 1 \end{pmatrix} \tilde{\boldsymbol{B}}_i$, we can derive the following equation:

$$\tilde{\boldsymbol{B}}_{i,cam1} = \begin{pmatrix} \boldsymbol{R}_1 & \boldsymbol{t}_1 \\ 0 & 1 \end{pmatrix} \tilde{\boldsymbol{B}}_{i,cam0} \tag{21}$$

where $\boldsymbol{t}_1 = -\boldsymbol{R}_1\boldsymbol{O}_1$. Once four or more such reconstructed pairs $(\tilde{\boldsymbol{B}}_{i,cam1}, \tilde{\boldsymbol{B}}_{i,cam0})$ are obtained, $\boldsymbol{R}_1$ and $\boldsymbol{O}_1$ can be calculated by:

$$\begin{pmatrix} \boldsymbol{R}_1 & \boldsymbol{t}_1 \\ 0 & 1 \end{pmatrix} = [\tilde{\boldsymbol{B}}_{i,cam1}][\tilde{\boldsymbol{B}}_{i,cam0}]^{-1} \tag{22}$$

$$\boldsymbol{O}_1 = -\boldsymbol{R}_1^{-1}\boldsymbol{t}_1. \tag{23}$$

If we use L as the unit for the measurement of relative positions, Equation (23) can be re-written as:

$$\boldsymbol{O}_1 = -\frac{1}{L}\boldsymbol{R}_1^{-1}\boldsymbol{t}_1. \tag{24}$$

This completes the calibration of extrinsic parameters.

Proposition 3: For a set of image points of a camera, if there exists an L that derives a $\boldsymbol{K}$, $\boldsymbol{R}$, and $\boldsymbol{O}$, then for any $L > 0$ the algorithm produces the same $\boldsymbol{K}$, $\boldsymbol{R}$, and $\boldsymbol{O}$. (See Appendix for proof.)

Proposition 3 means that using a set of images, the radius L of the globe does not need to be known in order to get the correct calibration result. In addition, if two different radii are assigned separately, the results are consistent. However, the accuracy of the measurement of L does affect the calibrated position in physical units, *e.g.,* centimeter or inch.

Compared to previous methods, one advantage of our algorithm is that only one image taken from each camera is sufficient for calibration of all the cameras, as long as enough intersections (minimally four) are visible simultaneously by any two adjacent cameras. Another advantage is that the correctness of the calibration result can be determined directly since the spherical points (including the sphere center) are reconstructed immediately. If the distances from each of those points to the sphere center all equal to L, then the points lie on a spherical surface. This reconstruction result validates the calibration result. However, one drawback of our algorithm is that the accuracy of the result strongly depends on the initial detection of the intersection points.

4 Experimental Results and Discussion

The proposed calibration algorithm is validated using both synthetic data and real images. As for synthetic data, a set of image points are generated by projecting a synthetic 3D globe model onto the image plane. To simplify the experimental process, the image of the globe center is assumed to be known. To

test our algorithm under a single-camera setup, we use the following camera property: $\alpha = 1200$, $\beta = 1000$, $s = 1$, $x_0 = 400$, and $y_0 = 300$. For the globe, the angle between each meridian is 15^o, $L = 150$, $\boldsymbol{A} = (0, 35, 1500)^T$. The experimental result computes exactly the same parameters as the assigned camera property. To test our algorithm under a two-camera setup, we use the following globe property: the angle between each meridian is 15^o, $L = 200$, $\boldsymbol{A} = (50, 10, 2000)^T$; and the following camera properties: for Camera_0 (the reference camera), $\alpha = 1000$, $\beta = 1000$, $s = 1$, $x_0 = 400$, and $y_0 = 400$; for Camera_1, $\alpha = 1000$, $\beta = 800$, $s = 0$, $x_0 = 320$, and $y_0 = 240$. The position and orientation of Camera_1 relative to Camera_0 is defined by the transformation in the following order: rotate -10^o about y-axis, rotate 10^o about x-axis, rotate 0^o about z-axis, translate by the vector $(200, 100, -500)^T = L(1, 0.5, -2.5)^T$. The experimental result computes exactly the same intrinsic and extrinsic parameters of the cameras.

Furthermore, the calibration algorithm for intrinsic parameters is also validated using real images. One image (image size: 2048×1536) used for testing is shown in Figure 3 (left) with twenty-two detected intersections marked as red circles and three fitted ellipses in blue. Here we employ the direct least-squares method [15] for ellipse fitting. Twenty-two intersection points are used mainly for ellipse fitting, but it is not necessary to use all of them to calculate the intrinsic parameters in the steps afterward. Table 1 gives the calibration results. Case one uses all the twenty-two points for calibration, while case two uses fourteen points as marked in Figure 3 (right). Unlike other methods [13] [14], which validate the calibration results by calculating the relative difference to the result obtained from the chessboard method [16], our algorithm directly reconstructs the spherical points; therefore, we use the relative error in the reconstruction result as the measure of the correctness of the calibration result. The RMSE is the root mean squared error measured on the reconstructed sphere. We define the error as the normalized difference between the distance from a reconstructed point to the reconstructed center and the actual radius length L, *i.e.*, $E = |\frac{\|\boldsymbol{B_i} - \boldsymbol{A}\|}{L} - 1|$, where $\boldsymbol{B_i}$'s are the reconstructed spherical points and $\boldsymbol{A}$ is the reconstructed sphere center. Then, the RMSE is calculated as $RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N} E^2}$, where N is the total number of intersection points used. The reconstruction errors are around 2% in both cases, which reflects the error in the calibration results. As mentioned earlier, the accuracy of our algorithm depends largely on the initial detection of the intersection points. The error in the calibration results is caused mainly by the inaccuracy in the detection step. In addition, the fact that our algorithm does not take radial distortion into account may also contribute to the error.

In the current stage, the initial intersection points are picked manually. We are still working on automatically detecting those points. One possible method is to cover the grids on the globe, which are formed by the longitude and latitude lines, with alternate black and white blocks, a pattern that resembles the chessboard. A globe painted in this way will facilitate the detection of the intersection points. Alternatively, simply putting a marker on each of those inter-

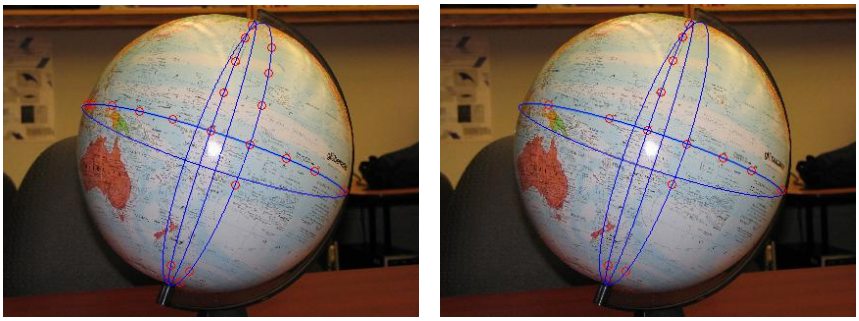

Fig. 3. Globe images with detected intersections and fitted ellipses.

Table 1. The calibration results with real data.

Cases	α	β	s	x_0	y_0	$RMSE*$	$\min(E)*$	$\max(E)$
One	4448.99	5261.07	823.664 (79.51°)	896.6	974.848	2.486	0.095	9.037
Two	4413.78	4931.82	439.022 (84.32°)	1061.76	1051.12	1.905	0.002	4.918

$*$: the unit for $RMSE$ and E is percentage (%).

Fig. 4. (Left and middle): different views of a globe captured by different cameras in a multi-camera set up (right).

sections may also help the automatic detection. We are also working on using the proposed approach to calibrate a multi-camera setup, as shown in Figure 4, to be used to capture video for a free-viewpoint 3D display.

5 Conclusion

We proposed a new multi-camera calibration algorithm using a simple household object, a globe. The approach uses a single view of the globe to calibrate a camera, and does not need to move the globe to multiple locations. The effectiveness of the algorithm is validated through experiments on both synthetic data and real images. In the future, we will test our algorithm with noisy data to verify its robustness.

References

1. Klette, R., Schlüns, K., Koschan, A.: Computer vision: three-dimensional data from images. first edn. Springer (1998)
2. Agrawal, M., Davis, L.S.: Camera calibration using spheres: A semi-definite programming approach. In: ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision. (2003) 782–789
3. Sturm, P., Maybank, S.: On plane-based camera calibration: A general algorithm, singularities, applications. In: CVPR '99: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 1. (1999) 432–437
4. Hammarstedt, P., Sturm, P., Heyden, A.: Degenerate cases and closed-form solutions for camera calibration with one-dimensional objects. In: ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision. (2005) 317–324
5. Hartley, R.I.: Self-calibration of stationary cameras. International Journal of Computer Vision **22**(1) (1997) 5–23
6. Horaud, R., Mohr, R., Lorecki, B.: Linear camera calibration. In: Proceedings of the IEEE International Conference on Robotics and Automation. Volume 2. (1992) 1539–1544
7. Basu, A., Ravi, K.: Active camera calibration using pan, tilt and roll. IEEE Transactions on Systems, Man and Cybernetics **27**(3) (1997) 559–566
8. Borghese, N.A., Colombo, F.M., Alzati, A.: Computing camera focal length by zooming a single point. Pattern Recognition **39**(8) (2006) 1522–1529
9. Barreto, J., Daniilidis, K.: Wide area multiple camera calibration and estimation of radial distortion. In: Proceedings of The Fifth Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras. (2004)
10. Caprile, B., Torre, V.: Using vanishing points for camera calibration. International Journal of Computer Vision **4**(2) (1990) 127–140
11. Daucher, N., Dhome, M., Lapresté, J.T.: Camera calibration from spheres images. In: ECCV '94: Proceedings of the Third European conference on Computer vision. (1994) 449–454
12. Zhang, H., Wong, K.Y.K., Zhang, G.: Camera calibration from images of spheres. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(3) (2007) 499–502
13. Zhang, Z.: Camera calibration with one-dimensional objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(7) (2004) 892–899
14. Wang, L., Wu, F.C., Hu, Z.Y.: Multi-camera calibration with one-dimensional object under general motions. In: ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision. (2007)
15. Fitzgibbon, A., Pilu, M., Fisher, R.B.: Direct least square fitting of ellipses. IEEE Transactions on Pattern Analysis and Machine Intelligence **21**(5) (1999) 476–480
16. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(11) (2000) 1330–1334

Appendix

Proposition 1: The motion constraint expressed in Equation (8) is equivalent to that in Equation (9).

Proof: For the ith camera, suppose its intrinsic parameters ($\boldsymbol{K}_i$) and extrinsic parameters ($\boldsymbol{R}_i$ and $\boldsymbol{t}_i$) are invariant when capturing the images. The vanishing point $\boldsymbol{v}_{ij}$ is the image of the infinite point $\boldsymbol{V}_{ij}$, therefore:

$$\forall j \; \tilde{\boldsymbol{v}}_{ij} = \boldsymbol{K}_i[\boldsymbol{R}_i \mid \boldsymbol{t}_i]\tilde{\boldsymbol{V}}_{ij}. \tag{25}$$

Let the world coordinate system coincide with the ith camera's coordinate system, then Equation (25) can be rewritten as:

$$\forall j \; \boldsymbol{v}_{ij} = \boldsymbol{K}_i\boldsymbol{V}_{ij}. \tag{26}$$

If Equation (8) is satisfied, *i.e.*, the motion is critical, then from Equation (26) the following equation is also satisfied:

$$\forall j \; \boldsymbol{v}_{ij}^T\omega'\boldsymbol{v}_{ij} = \boldsymbol{V}_{ij}^T\boldsymbol{K}_i^T\omega'\boldsymbol{K}_i\boldsymbol{V}_{ij} = \boldsymbol{V}_{ij}^T\omega''\boldsymbol{V}_{ij} = 0. \tag{27}$$

This is exactly the same expression as Equation (9). Similarly, one can prove that if Equation (9) is satisfied, then Equation (8) is also satisfied.□

Proposition 2: For a set of image points of a camera, if there exists an L that can be used to derive a $\boldsymbol{K}$, then for any $L > 0$ the algorithm produces the same $\boldsymbol{K}$.

Proof: Suppose L is scaled by a factor $\gamma > 0$, *i.e.*, the radius of the globe used for calculation becomes $L' = \gamma L$. Note that the assignment of the radius does not affect the image points. From Equation (7), the current $\boldsymbol{x}'$=$\gamma^2\boldsymbol{x}$, *i.e.*, $x'_i = \gamma^2 x_i$ for $i \in [1, 6]$. Therefore, $y'_0 = y_0$, $z'_A = \gamma z_A$, $\alpha' = \alpha$, $\beta' = \beta$, $s' = s$, and $x'_0 = x_0$ using Equations (10) to (15). Hence, $\boldsymbol{K}' = \boldsymbol{K}$.□

Proposition 3: For a set of image points of a camera, if there exists an L that derives a $\boldsymbol{K}$, $\boldsymbol{R}$, and $\boldsymbol{O}$, then for any $L > 0$ the algorithm produces the same $\boldsymbol{K}$, $\boldsymbol{R}$, and $\boldsymbol{O}$.

Proof: We consider two cases: 1) the camera is the reference camera; and 2) the camera is not the reference camera.

Case 1: Because $\boldsymbol{R} = \boldsymbol{I}$ and $\boldsymbol{O} = 0$, the proposition can be proved directly from Proposition 2.

Case 2: Suppose L is scaled by a factor $\gamma > 0$. From Proposition 2, we know that as a consequence z_A is scaled by γ, but K remains the same. z_{B_i} becomes $z'_{B_i} = \gamma z_{B_i}$ from Equation (19). Therefore, $\boldsymbol{B'}_{i,cam0} = \gamma\boldsymbol{B}_{i,cam0} = \gamma\boldsymbol{I}(\boldsymbol{B}_i - 0) = \gamma\boldsymbol{B}_i$ and $\boldsymbol{B'}_{i,cam1} = \gamma\boldsymbol{B}_{i,cam1} = \gamma\boldsymbol{R}_1(\boldsymbol{B}_i - \boldsymbol{O}_1)$ from Equation (20). Therefore, $\boldsymbol{B'}_{i,cam1} = \boldsymbol{R}_1(\boldsymbol{B'}_{i,cam0} - \gamma\boldsymbol{O}_1)$. Using homogeneous representation, $\tilde{\boldsymbol{B}}'_{i,cam1} = \begin{pmatrix} \boldsymbol{R}_1 & \boldsymbol{t'}_1 \\ 0 & 1 \end{pmatrix} \tilde{\boldsymbol{B}}'_{i,cam0}$, where $\boldsymbol{t'}_1 = \gamma\boldsymbol{t}_1$. The calibration result of $\boldsymbol{R}_1$ is not affected by the scale factor γ using Equation (22). Using Equation (24), the calibration result of $\boldsymbol{O}_1$ is also not affected, as γ is eliminated from both the numerator and the denominator. Hence, the calibration result is independent of the assignment of L.□