

# On Calibrating a Camera Network Using Parabolic Trajectories of a Bouncing Ball

Kuan-Wen Chen<sup>1</sup>, Yi-Ping Hung<sup>1,2</sup>, Yong-Sheng Chen<sup>3</sup>

<sup>1</sup>*Department of Computer Science and Information Engineering,  
National Taiwan University, Taipei, Taiwan*

<sup>2</sup>*Graduate Institute of Networking and Multimedia,  
National Taiwan University, Taipei, Taiwan*

<sup>3</sup>*Department of Computer Science,  
National Chiao Tung University, Hsinchu, Taiwan*

*E-mail: r93922014@ntu.edu.tw, hung@csie.ntu.edu.tw, ysc@csie.nctu.edu.tw*

## Abstract

*Camera networks are often used in visual surveillance systems for wide-range monitoring. In this paper, we present a novel method for calibrating a camera network, which uses the trajectory of a bouncing ball as the calibration data. An important feature of our method is the use of the parabolic property of a ball's bouncing trajectory. This parabolic trajectory will lie on a plane, called the parabolic trajectory plane (PT-plane), so that the relationship between the trajectory's points and their corresponding image points is a homography. Combining the vertical velocity determined by the earth's gravity and the horizontal velocity calculated from the homography, we can compute the 2D coordinates of the trajectory points on the PT-plane. By throwing the ball multiple times, we obtain calibration points on multiple planes for calibrating both intrinsic and extrinsic parameters of the networked cameras. Experimental results have demonstrated the feasibility and accuracy of the proposed method.*

## 1. Introduction

Camera networks are extensively used in visual surveillance systems because they can monitor the activities of targets over a larger area than a single camera. If a camera network has been calibrated, it can achieve cooperative tracking and provide more 3D information. Hence, how to calibrate a camera network is an important task for visual surveillance systems.

Camera calibration is a widely used technique in 3D computer vision. There are many methods in the

literature. According to the dimensionality of the calibration objects, we can classify those methods into four main categories:

- **3D calibration object.** Camera calibration is performed by using a 3D calibration object whose geometry in 3D space is known beforehand with good precision. For example, the special property of house corners [4], a cuboid structure seen in the camera image (or more generally, a parallelepiped structure) [3], or a plane undergoing a precisely known translation [13] can be used as the reference object to infer camera parameters. While this approach can be performed very efficiently, it always requires an expensive setup before calibration.
- **2D calibration object.** Some 2D plane based calibration methods only need to observe a planar pattern from a few different view points. Maybank's [12] and Zhang's [14] methods belong to this category. The motion of the plane can be unknown, and the setup is very easy.
- **1D calibration object.** This approach just uses a 1D object, but requires one endpoint of the stick to be fixed [15].
- **Self-calibration (also called using 0D features).** Techniques in this category do not need any known calibration objects. Only the image point correspondences are required. By moving a camera in a static scene, the rigidity of the scene provides two constraints [9, 10] for determining the camera parameters. Although no calibration objects are needed, it is difficult to obtain accurate and robust results.

Relative camera poses have to be determined to obtain the geometric relationship in a camera network. Usually we can use known 3D positions of objects in the world coordinate system to calculate the extrinsic parameters in closed form [6, 2]. Otherwise, we can use an iteration method, such as Church's method, to solve this problem [5, 7]. However, using a fixed calibration object to calibrate a camera network always has some difficulties. For example, the calibration objects need to be seen by more than one camera, therefore the fixed calibration objects need to be large enough. Large calibration objects are not always easy to use and to take along, or to anticipate the proper size in advance.

In this paper, we proposed a new calibration method which uses a 0D object, a ball. When throwing a bouncing ball, gravity causes a parabolic trajectory, and we can use this property to calibrate a camera network. A ball is much easier to take along, and for the purpose of being seen by two or more cameras we just have to throw a ball to produce a proper trajectory size. In this paper, we will discuss three cases for a camera network: the intrinsic parameters of all cameras are known, the intrinsic parameter of only one camera is known, and the intrinsic parameters of all cameras are unknown. We show how to use a bouncing ball to perform camera calibration in each case.

The main idea of this work is to utilize gravity to estimate camera parameters. The only similar work that we found is the one done by Sturm and Quan [11]. However, their method is different from ours in that they first estimate the infinite homography [8] between two captured images by using corresponding vanishing points and lines and then obtain intrinsic parameters and relative pose from the estimated infinite homography. Our method is more flexible in the following three aspects. First, our method can be applied even if there is only one camera, while Sturm and Quan's method needs to use at least two cameras. Second, our method can be used to estimate all the intrinsic parameters, while Sturm and Quan's method can estimate only a subset of the intrinsic parameters due to the insufficient constraints provided by the infinite homography. Third, Sturm and Quan's method will suffer from the singularity problem when the optical axes of two involved cameras are parallel, while our method does not have this problem and is therefore more flexible.

This paper is organized as follows. In Section 2, we describe the basic constraints from observing a trajectory plane. In Section 3, we describe how to calibrate a camera network using a bouncing ball in three cases. Section 4 provides the experimental results, using computer simulation to validate the proposed technique. Finally, we conclude the paper in Section 5.

## 2. Basic Equations

Because the parabolic trajectory of a bouncing ball must be on one plane, called the parabolic trajectory plane (PT-plane), we can examine the constraints of the camera's intrinsic parameters by observing that plane. We start with the notation used in this paper.

### 2.1. Notation

A 2D point in an image is denoted by  $m=[u,v]^T$ , and a 3D point in world coordinate system is denoted by  $M=[X,Y,Z]^T$ . We use  $\tilde{x}$  to denote the homogeneous coordinates augmented vector by adding 1 at the last element:  $\tilde{m}=[u,v,1]^T$  and  $\tilde{M}=[X,Y,Z,1]^T$ . Using the pinhole camera model the projective geometry of 2D and 3D points is given by

$$s\tilde{m} = A[R|T]\tilde{M}, \text{ with } A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where  $s$  is an arbitrary scale factor, and  $[R|T]$  is the extrinsic parameters comprising the rotation and translation matrix related from the world coordinates system to the camera coordinates system. In the intrinsic parameter matrix  $A$ ,  $\alpha$  and  $\beta$  defines the scale factors in image  $u$  and  $v$  axes,  $\gamma$  describes the skewness of the two axes, and  $u_0, v_0$  are the coordinates of the principal point. In the following, we use the abbreviation  $A^{-T}$  for  $(A^{-1})^T$  or  $(A^T)^{-1}$ .

### 2.2. Homography between PT-Plane and Image Plane

When we arbitrarily throw a bouncing ball, we can use the camera to capture a sequence of frames. Because of gravity the trajectory of the ball must be a parabola and on one plane, as shown in Figure 1. Image points of the parabolic trajectory are then identified in the sequence of frames. These image points can then be used in the following calibration procedure.

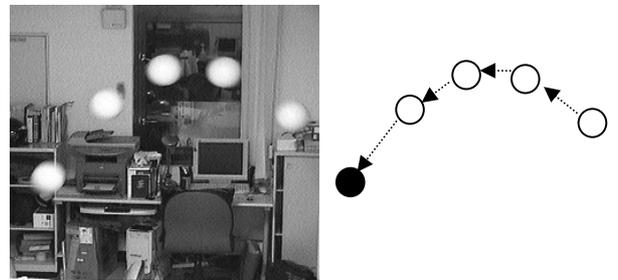


Figure 1. A parabolic trajectory of a bouncing ball

Without loss of generality, we assume that the PT-plane is on  $Z = 0$  of the world coordinate system and denote the  $i$ th column of the rotation matrix  $R$  by  $r_i$ . From (1), we have

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \quad (2)$$

Because  $Z$  is always equal to zero, then  $\tilde{M} = [X, Y, 1]^T$ . Therefore, a point  $M$  on the PT-plane and its point  $m$  in image is related by a homography  $H$ :

$$s\tilde{m} = H\tilde{M} \quad \text{with } H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}. \quad (3)$$

We set the  $X$  axis on the horizontal, set the  $Y$  axis on the vertical, denote the  $i$ th frame's capture time by  $T_i$ , denote the horizontal velocity of the bouncing ball by  $V$  which is unknown previously, and assume we observe the highest point of the parabola, denoting its capture time by  $T_h$ . Because of the properties of the parabola, every point  $[X, Y]^T$  in the PT-plane can be represented by

$$X = V(T_h - T_i), \quad Y = \frac{1}{2}g(T_h - T_i)^2, \quad (4)$$

where  $g$  is the earth's gravity equal to  $9.8 \text{ ms}^{-2}$ . Then we denote  $H = [h_1 \ h_2 \ h_3]$ . By substituting (4) into (2) and (3), we have

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [h_1 \ h_2 \ h_3] \begin{bmatrix} V(T_h - T_i) \\ \frac{1}{2}g(T_h - T_i)^2 \\ 1 \end{bmatrix} = [Vh_1 \ h_2 \ h_3] \begin{bmatrix} (T_h - T_i) \\ \frac{1}{2}g(T_h - T_i)^2 \\ 1 \end{bmatrix}. \quad (5)$$

Then we denote  $H' = [Vh_1 \ h_2 \ h_3] = [h_1' \ h_2 \ h_3]$  and  $X' = (T_h - T_i)$  and substitute them into (5). We have

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H' \begin{bmatrix} X' \\ Y \\ 1 \end{bmatrix}. \quad (6)$$

Clearly, the  $3 \times 3$  matrix of the  $H'$  is defined up to a scale factor.

### 2.3. Constraints on Intrinsic Parameters

In equation (6),  $Y$ , and  $X'$  are known and  $\tilde{m} = [u, v, 1]^T$  can be observed from the image. We can estimate the  $H'$  by using the estimation method of homography between the model plane and its image, as in [14]. From (3) and  $H = [h_1 \ h_2 \ h_3]$ , we have

$$[h_1 \ h_2 \ h_3] = \lambda A [r_1 \ r_2 \ t], \quad (7)$$

where  $\lambda$  is an arbitrary scalar. According the knowledge that  $r_1$  and  $r_2$  are orthonormal, we have

$$r_1 \cdot r_2 = 0, \quad (8)$$

$$|r_1|^2 = |r_2|^2. \quad (9)$$

Substituting (7) into (8) and (9), we have the following two constraints:  $A$

$$h_1^T A^{-T} A^{-1} h_2 = 0, \quad (10)$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2. \quad (11)$$

Furthermore, from the equations  $[Vh_1 \ h_2 \ h_3] = [h_1' \ h_2 \ h_3]$ , (10), and (11), we can substitute  $h_1'$  for  $h_1$  and derive

$$\left(\frac{h_1'}{V}\right)^T A^{-T} A^{-1} h_2 = 0,$$

$$\left(\frac{h_1'}{V}\right)^T A^{-T} A^{-1} \left(\frac{h_1'}{V}\right) = h_2^T A^{-T} A^{-1} h_2.$$

In turn, we obtain

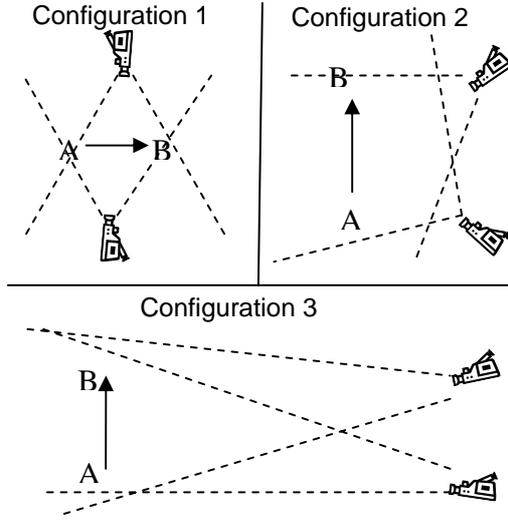
$$(h_1')^T A^{-T} A^{-1} h_2 = 0, \quad (12)$$

$$V = \sqrt{\frac{(h_1')^T A^{-T} A^{-1} (h_1')}{h_2^T A^{-T} A^{-1} h_2}}. \quad (13)$$

Therefore, when we throw a ball one time, we can estimate an  $H'$ . There are two basic constraints: one for the intrinsic parameters and the other for calculating the horizontal velocity of the bouncing ball.

## 3. Calibrating a Camera Network Using a Bouncing Ball

In this paper, we consider calibration of a camera network consisting of one or more than one cameras. Without loss of generality, we specify the coordinate system of the first camera to be the global coordinate system. Figure 2 illustrates three common configurations of a camera network comprising two cameras. In the first configuration, for example, we need a calibration object which can be seen concurrently by two cameras pointing in the opposite directions. Obviously, 3D calibration objects and 2D calibration planes are difficult to be used, while 1D calibration object and the bouncing ball used in this work are more suitable in this configuration. In the second configuration a small calibration object can be used because the overlapped area is near the cameras. In the third configuration, however, a large object may be required due to the far overlapped area. Hence, it is hard to find a general calibration object suitable for many camera network configurations. In our method, we arbitrarily throw a ball from point A to point B, as shown in Figure 2, with a proper trajectory size which can be captured by all cameras. Then we can use this trajectory to calibrate the camera network. In the following, we discuss three cases and propose how to calibrate the camera network in each case.



**Figure 2. Three common configurations of a camera network and a sample trajectory of a bouncing ball thrown from A to B**

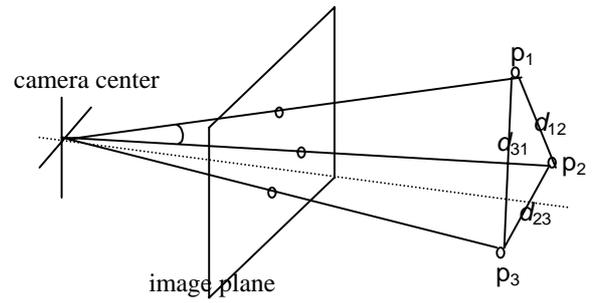
### 3.1 Case 1: Intrinsic Parameters of All Cameras Are Known

The first case is that we assume all intrinsic parameters of the cameras in the camera network are known beforehand. This assumption is reasonable because in many visual surveillance systems the cameras' intrinsic parameters have been calibrated before being installed. Here we use two cameras, denoted by Cam1 and Cam2, to demonstrate the calibration procedure for extrinsic parameters of a camera network. The calibration procedure can be easily applied to configurations with more than two cameras.

The proposed calibration procedure consists of three steps. First, we arbitrarily throw a ball one time in the overlapped area, like in Figure 2, where both cameras can easily capture the trajectory of the ball. As presented in the previous section, we can estimate  $H_1'$  for Cam1 and  $H_2'$  for Cam2 according to the trajectories captured by each camera. Because the intrinsic parameters of two cameras, denoted by  $A_1$  for Cam1 and  $A_2$  for Cam2, are known and  $H_1'$  and  $H_2'$  can be estimated, we can calculate the  $V$  of the bouncing ball by using the equation (13). Theoretically, we will get the same  $V$  for each camera. However, noise always exists thus the estimated value of  $V$  differs for different cameras. Averaging is a simple method to compromise with the discrepancy.

Second, without loss of generality, we assume the parabolic trajectory is on the plane  $Z = 0$  of the world coordinate system. Each point of the trajectory can then be represented by  $[X, Y, 0]^T$ . The coordinates of all points on the trajectory can be calculated by (4), after knowing

the horizontal velocity of the bouncing ball  $V$  in the first step. We choose three of these points and compute the distances between each pair of these three points. The three points may be chosen arbitrarily. However, in our research we choose the first, the middle, and the last points because the distances between them are usually larger thus tend to be less sensitive to noise. These three points project on each camera, as shown in Figure 3. Because we have the intrinsic parameters of each camera and the distances between each pair of three points, we use Church's method [5] to calculate the 3D coordinates of feature points  $p_1$ ,  $p_2$ , and  $p_3$  in the camera coordinate system.



**Figure 3. Three points project to an image plane**

The goal of third step is to estimate the extrinsic parameters between Cam1 and Cam2, that is, the rotation and translation matrix  $[R|T]$  from the coordinate system of Cam2 to the coordinate system of Cam1. We specify the coordinate system of Cam1 as the global coordinate system. By using the method described in the second step, we calculate the coordinate pair,  $v_1 = [X_1, Y_1, Z_1]^T$  in the coordinate system of Cam1 and  $v_2 = [X_2, Y_2, Z_2]^T$  in the coordinate system of Cam2, for each of the three feature points. From each coordinate pair, we have

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = [R|T] \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix}.$$

The  $[R|T]$  is the extrinsic parameter from Cam2 to Cam1, which can be calculated through the least-squares solution of  $[R|T]$  based on SVD, as proposed by Arun et al. [1].

### 3.2 Case 2: Intrinsic Parameters of Only One Camera Is Known

In the second case, only one of the cameras in the camera network is calibrated in advance. Our basic idea is to calibrate the other cameras using the calibrated camera. In this case, we also consider a two-camera system. We denote the calibrated camera by Cam1 and the other

camera by Cam2. Because Cam1 has been calibrated, the intrinsic parameters are known, hence, we can estimate the horizontal velocity  $V$  and the coordinates in the coordinate system of Cam1 for a trajectory by using the proposed method in Section 3.1. By throwing the ball multiple times, we have calibration points on multiple planes with known 3D coordinates in the coordinate system of Cam1. Both the intrinsic and extrinsic parameters of all other cameras can be calibrated by using these calibration points through conventional camera calibration procedure [13]. Another way of calibration is to use the value of  $V$  to calculate the planar coordinates for each trajectory. Then, the feature points on multiple 2D planes can be used to calibrate the cameras with Zhang's method [14].

### 3.3 Case 3: Intrinsic Parameters of All Cameras Are Unknown

In this case, two methods can be used to calibrate the cameras. With the first method, we temporarily employ an auxiliary calibrated camera to observe the bouncing ball. The case is now reduced to the second case, and all the cameras in the network can be calibrated by using the proposed method in Section 3.2.

If there is no calibrated camera at hand, we still can calibrate the camera network by using the following closed-form solution. Because the trajectory of a bouncing ball is parabolic and it must lie on one plane, each trajectory can provide a constraint, the equation (12), on the intrinsic parameters, according to our derivation in Section 2. Let

$$B = A^{-T} A^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}. \quad (14)$$

Note that  $B$  is symmetric, so it can be represented by a 6D vector

$$b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T. \quad (15)$$

Let the  $i$ th column vector of  $H$  be  $h_i' = [h_{i1}, h_{i2}, h_{i3}]^T$ . From the constraint (12), we have

$$h_i'^T B h_i' = w^T b = 0, \quad (16)$$

with  $w =$

$$[h_{11} h_{21}, h_{11} h_{22} + h_{12} h_{21}, h_{12} h_{22}, h_{13} h_{21} + h_{11} h_{23}, h_{13} h_{22} + h_{12} h_{23}, h_{13} h_{23}]^T.$$

If we throw the ball  $n$  times,  $n$  trajectories are observed. By stacking  $n$  such equation as (16), we have

$$Wb = 0, \quad (17)$$

where  $W$  is a  $n \times 6$  matrix. Obviously, if  $n \geq 5$ , we have a closed-form solution  $b$  up to a scale factor. The well-known solution to (17) is the eigenvector of  $W^T W$  associated with the smallest eigenvalue. After estimating the vector  $b$ , we can compute all camera intrinsic parameters as follows.

$$v_0 = (B_{12} B_{13} - B_{11} B_{23}) / (B_{11} B_{22} - B_{12}^2)$$

$$\lambda = B_{33} - [B_{13}^2 + v_0 (B_{12} B_{13} - B_{11} B_{23})] / B_{11}$$

$$\alpha = \sqrt{\lambda / B_{11}}$$

$$\beta = \sqrt{\lambda B_{11} / (B_{11} B_{22} - B_{12}^2)}$$

$$\gamma = -B_{12} \alpha^2 \beta / \lambda$$

$$u_0 = v_0 / \beta - B_{13} \alpha^2 / \gamma$$

In practice, the solution of equation (17) may cause trouble when the signs of  $B_{11}$  and  $\lambda$  are not the same. Hence we use an optimization procedure to calculate the intrinsic parameters. We set the initial value of  $\gamma$  to be 0, and set the initial values of  $(u_0, v_0)$  to be the image center. Then equation (17) reduces to non-homogeneous equation and the values of  $\alpha$  and  $\beta$  can be easily calculated. The intrinsic parameters are then refined iteratively.

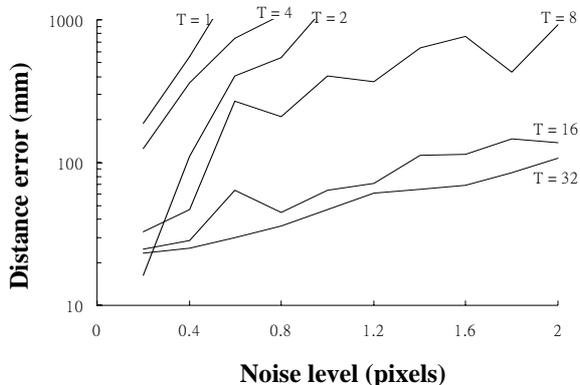
Once the intrinsic parameters of all cameras are known, this case reduces to the first one and the camera network can be calibrated with the method in Section 3.1.

## 4. Experimental Results

The proposed methods have been evaluated by using computer simulation. In our experiment, the simulated trajectories are with different angles and random horizontal velocities  $V$  ranging from 1 to 3 m/s.

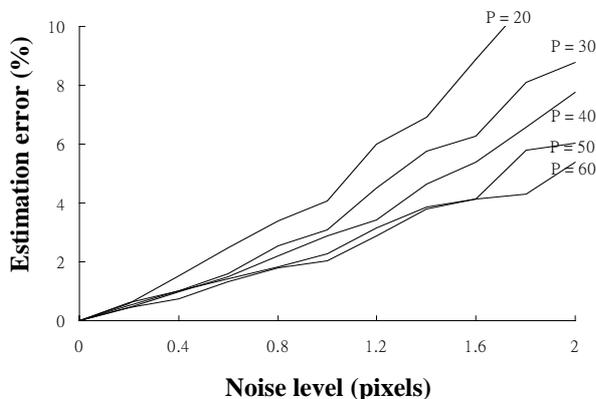
Consider the first case in Section 3.1 where two cameras were mounted in the third configuration, as shown in Figure 2. Each time the ball was thrown, each of the two cameras captured 25 frames. That is, we observed 25 points from each parabolic trajectory. In practice we often can observe more than 25 points. Gaussian noise with zero mean and standard deviation  $\sigma$  was added to the projected image points. The calibration procedures in Section 3.1 were used to compute  $[R|T]$ . To assess the accuracy of the estimated  $[R|T]$ , we arbitrarily chose a 3D point in the global coordinate system and then projected it to these two cameras. By using the projected image points of two cameras and the estimated  $[R|T]$ , we computed the 3D coordinate via triangulation and calculated deviation from the ground truth. For each noise level varying from 0.2 pixels to 2 pixels, we conducted 100 independent trials and then averaged the results. In addition, we also evaluated the performance of the proposed method with respect to the number of trajectories ranging from 1 to 16. The results are illustrated in Figure 4. We can observe easily that errors

increase with the noise level and decrease with the number of trajectories.



**Figure 4. Distance errors vs. the noise level of the image points (T is the number of trajectories).**

Next, we consider the camera network calibration when the intrinsic parameters of all cameras are unknown without using an auxiliary calibrated camera. Consider a camera with the ground-truth intrinsic parameters:  $\alpha = 1000$ ,  $\beta = 1000$ ,  $\gamma = 0$ ,  $u_0 = 319.5$ ,  $v_0 = 239.5$ . The image resolution is  $640 \times 480$ . We arbitrarily produced 16 different simulated trajectories that were projected in the field of view of the camera. As in the previous experiment, Gaussian noise with zero mean and standard deviation  $\sigma$  was added to the projected image points. For each noise level ranging from 0.2 pixels to 2 pixels, we conducted 100 independent trials and then averaged the results. Figure 5 illustrates the average deviation of estimated  $\alpha$  and  $\beta$  from their ground-truth values. To evaluate the performance of the proposed method with respect to the number of points in each trajectory, we varied the amount of captured points from 20 to 60, which is common in practice when the frame rate of the camera is 30Hz. As we can see from Figure 5, errors increase proportionally with the noise level and decrease with the number of captured points.



**Figure 5. Average estimation error of intrinsic parameters  $\alpha$  and  $\beta$  vs. the noise level of the image points (P denotes number of captured points)**

According to the results of these two experiments, we can obtain more accurate camera parameters either by increasing the number of times the ball is thrown or by increasing the amount of captured points for each trajectory.

## 5. Conclusions

In this paper, we proposed a new camera calibration method using a bouncing ball. Also, we presented how to apply this novel method to calibrate a camera network. We considered three different cases encountered in calibrating a camera network, where the intrinsic parameters of all, only one, or none of the cameras are known. Calibration procedure for each case was discussed. Experimental results have demonstrated the feasibility and accuracy of the proposed method.

Camera calibration has been studied extensively in computer vision and photogrammetry. The existing techniques include those using 3D objects, 2D objects, and 1D calibration objects. To our knowledge, there are no calibration methods really use 0D objects. Self-calibration is sometimes called “using 0D features”. However, self-calibration uses image point correspondences, which is different from using 0D objects. Therefore, this proposed calibration technique using 0D objects fills a missing dimension. Besides, calibration of a camera network is often not easy because of the fixed size of calibration objects. The proposed technique is practical, especially for calibrating a camera network, because we can simply throw a ball in the overlapped area of the cameras.

## Acknowledgements

This work was supported in part by the Ministry of Economic Affairs, Taiwan, under Grant 94-EC-17-A-02-S1-032.

## References

- [1] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-Squares Fitting of Two 3-D Point Sets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, Sep. 1987, pp.698-700.
- [2] P. J. Besl, “A Method for Registration of 3D Shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, Feb, 1992, pp.239-256.

- [3] C.-S. Chen, C.-K. Yu, and Y.-P. Hung, "New Calibration-free Approach for Augmented Reality Based on Parameterized Cuboid Structure," *Proceedings of the IEEE International Conference on Computer Vision*, Corfu, Greece, 1999, pp.30-37.
- [4] H. L. Chou and W. H. Tsai, "A new approach to robot location by house corners," *Pattern Recognition*, Vol. 19, No. 6, 1986, pp.439-451.
- [5] E. Church, "Revised Geometry of the Aerial Photograph," *Bulletin of Aerial Photogrammetry*, No. 15, 1945.
- [6] D. DeMenthon and L. S. Davis, "Exact and Approximate Solutions of the Perspective-Three-Point Problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 11, Nov, 1992, pp.1100-1105.
- [7] M. A. Fischler and R. C. Bolles, "Random Sample consensus: A Paradigm for Model fitting with Applications to image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp.381-395.
- [8] O. Faugeras, "Stratification of Three-Dimensional Vision: Projective, Affine, and Metric Representations," *Journal of the Optical Society of America A*, Vol. 12, No. 3, Mar, 1995, pp.465-484.
- [9] Q.-T. Luong, "Matrice Fondamentale et Calibration Visuelle sur l'Environment-Vers une plus grande autonomie des system robotiques," PhD thesis, Universite de Paris-Sud, Centre d'Orsay, Dec, 1992.
- [10] S. J. Maybank and O. D. Faugeras, "A theory of self-calibration of a moving camera," *The International Journal of Computer Vision*, Vol. 8, No. 2, Aug, 1992, pp.123-152.
- [11] P. Sturm and L. Quan, "Camera Calibration and Relative Pose Estimation from Gravity," *International Conference on Pattern Recognition*, Barcelona, Spain, Vol. 1, No. 1, Sep, 2000, pp.72-75.
- [12] P. Sturm and S. Maybank, "On Plane-based Camera Calibration: A General Algorithm, Singularities, Applications," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognitions*, Fort Collins, Colorado, USA, Jun, 1999. IEEE Computer Society Press. pp.432-437.
- [13] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv camera and lenses," *IEEE Journal of Robotics and Automation*, Vol. 3, No. 4, Aug, 1987, pp.323-344.
- [14] Z.-Y. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, Nov, 2000, pp.1330-1334.
- [15] Z.-Y. Zhang, "Camera Calibration with One-Dimensional Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 7, Jul, 2004, pp.892-899.