Camera Calibration by a Single Image of Balls: From Conics to the Absolute Conic

Hirohisa Teramoto and Gang Xu
Computer Vision Laboratory, Computer Science Department
Ritsumeikan University
Kusatsu, Shiga 525-8577, Japan

Abstract

In this paper we propose a new flexible technique to calibrate camera's intrinsic parameters from a single image of balls. Balls are projected onto the image as ellipses, or conics. Each conic provides two constraints on the intrinsic parameters. To estimate all five intrinsic parameters, we need three balls. Ball's size is arbitrary. Everyone can find a few balls around. It is more flexible than other predesigned calibration objects. Only one image is used, and there is no correspondence problem involved. We propose an algorithm to estimate the camera intrinsic parameters and the relative size and position of each ball optimally and simultaneously from the boundary points of each ball. Experimental results on both synthetic and real images show that the algorithm is effective and robust.

1 Introduction

Camera calibration is necessary if one is to obtain metric information from images. For example, we need focal length and principal point of each camera in order to compute 3D coordinates from a pair of matched images. Many approaches have been proposed to solve the problem. They are classified into two categories.

Approaches in the first category do the job by observing a predesigned calibration object whose geometry is known precisely [4, 15, 17]. The object usually consists of two or three planes orthogonal to each other, or equivalently, vanishing points for orthogonal directions [9]. These approaches require a calibration apparatus, or are limited to cases where orthogonal directions can be found. Recently, people have proposed to use planes [14, 7] or planar patterns [19, 13]. Certainly, a plane or planar pattern is more flexible and easier to prepare than previous calibration objects. But in this case, multiple images have to be used and the correspondence problem has to be solved.

The other category is called self-calibration, comprising of approaches that do not need any known geometry, but only require a certain number of image points be matched over two or more images [10, 8, 5, 12]. A pair of images provides two constraints over the camera paramters, in the form of Kruppa's equations. If enough images are obtained, the intrinsic parameters can be solved for. The problem with this category of approaches, however, is that the result is sensitive to noise. It is not yet mature enough for accurate calibration [2].

In this paper, we propose a new approach that uses balls as the calibration object. This falls in the first category, but it is easier for the users as balls are ubiquitous. Any balls, be it a soccer ball, a pingpong ball, a toy ball, can serve the purpose. Since only one image is needed, there is no correspondence problem here.

It is well known that a ball's image is a conic, and more concretely, an ellipse in practice. It is also known that the camera intrinsic parameters can be represented as the image of the *absolute conic*. We will show in this paper an algorithm to optimally compute the camera intrinsic parameters and the relative size and position of each ball from the images of balls. Experimental results with both synthetic and real images show that this approach is easy to use, effective and robust.

In the literature, balls have been first used for calibration by Penna [11]. But he only tried to compute the aspect ratio of the two image axes. In an attempt to calibrate cameras by vanishing lines, Beardsley *et al.* show that by rotating objects which have parallel lines, the trajectories of vanishing points are ellipses [1]. They found that the major axes of the ellipses intersect at the principal point. Based on this observation, they further propose to first determine the aspect ratio using three ellipses, then determine the principal point and finally determine the focal length. In a separate effort towards camera calibration, Daucher *et al.* propose to use "spheres" [3]. They obtained similar results as Beardsley *et al.* Compared with these approaches, our approach not only provides a general mathematical formulation of the problem which links conics with the concept of absolute conic, but also provides an efficient algorithm to solve for the intrinsic paramters, and the relative positions and sizes of the balls, in an optimal manner.

2 Images of Balls: Conics

Balls, seen from any directions, are round. The boundary is always a circle. The circle and the camera's focal point forms a circular cone, whose intersection with the image plane becomes an ellipse (it becomes a parabole if the ball intersects the image plane.)

As illustrated in Fig.1, we define the origin of the coordinate system for the circular cone to be the same as the camera's focal point, and the Z-axis is from the origin to the center of the circle, or equivalently, the center of the ball. Between the camera's coordinate system and that of the cone, there is no translation but only a rotation.

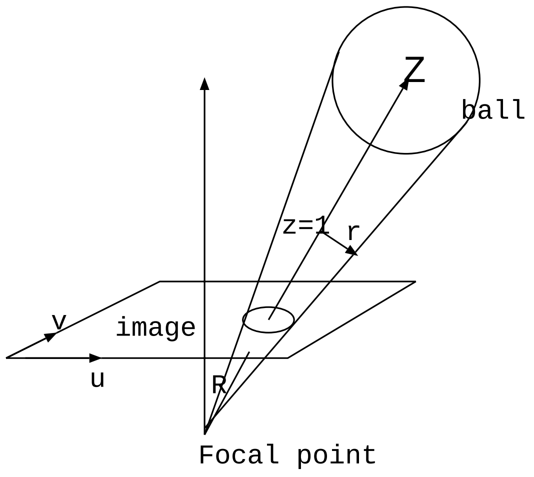

Figure 1: A ball is projected onto the image as an ellipse, or a conic.

We assume that the camera is a pinhole. The directly available pixel image coordinates $\mathbf{m} = [u, v]^T$ and the normalized image coordinates $\mathbf{x} = [x, y]^T$ are related by the following equation

$$\tilde{\mathbf{m}} = \mathbf{A}\tilde{\mathbf{x}} \, , \tag{1}$$

where $\tilde{\mathbf{m}} = [u, v, 1]^T$ and $\tilde{\mathbf{x}} = [x, y, 1]^T$ are the homogeneous coordinates of the point in pixel coordinates and normalized image coordinates, respectively, and $\mathbf{A}$ is the intrinsic matrix, given by

$$\mathbf{A} = \begin{bmatrix} \alpha_u & b & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \, , \tag{2}$$

with (u_0, v_0) the coordinates of the principal point in the pixel coordinate system, α_u, α_v the focal lengths for the u, v axes, and b the skew parameter between u and v axes.

Generally, a point $\mathtt{M} = [X, Y, Z]^T$ in the world coordinate system is projected onto the image point $\mathbf{m}$ by

$$s\tilde{\mathbf{m}} = \mathbf{A}\,[\,\mathbf{R} \quad \mathbf{t}\,]\,\tilde{\mathtt{M}} \, , \tag{3}$$

where s is a scalar, $\tilde{\mathtt{M}} = [X, Y, Z, 1]^T$ is the homogeneous coordinates of point $[X, Y, Z, 1]^T$, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector, respectively, between the world coordinate system and the camera coordinate system.

Since there is no translation here, we have $\mathbf{t} = \mathbf{0}$. Thus, a point $[X, Y, Z]^T$ on the cone is projected onto the image by

$$s\tilde{\mathbf{m}} = \mathbf{A}\mathbf{R}\mathtt{M} \, . \tag{4}$$

Moving $\mathbf{R}$ and $\mathbf{A}$ to the left side, we have

$$s\mathbf{R}^T\mathbf{A}^{-1}\mathbf{m} = \mathtt{M} \, . \tag{5}$$

For the third components of both sides to be equal, s must be

$$s = \frac{Z}{\mathbf{r}^T\mathbf{A}^{-1}\tilde{\mathbf{m}}} \, , \tag{6}$$

where $\mathbf{r}$ is the 3rd column vector of $\mathbf{R}$. It is not hard to see that $\mathbf{r}$ is actually the central axis of the circular cone in the camera coordinate system. Knowing $\mathbf{r}$ means knowing the ball's position up to depth. Assume that the circle has radius r at $Z = 1$. Multiplying the transpose of each side of (5) from the left yields

$$\tilde{\mathbf{m}}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\tilde{\mathbf{m}} = \frac{1}{s^2}\mathtt{M}^T\mathtt{M} = \frac{1}{s^2}(r^2+1)Z^2 \, .$$

Substituting (6) for the above equation yields

$$\tilde{\mathbf{m}}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\tilde{\mathbf{m}} = (r^2+1)\tilde{\mathbf{m}}^T\mathbf{A}^{-T}\mathbf{r}\mathbf{r}^T\mathbf{A}^{-1}\tilde{\mathbf{m}} \, . \tag{7}$$

Moving the right side to the left side, we have

$$\tilde{\mathbf{m}}^T\mathbf{Q}\tilde{\mathbf{m}} = 0 \, ,$$

where

$$\mathbf{Q} \cong \mathbf{A}^{-T}\mathbf{A}^{-1} - (r^2+1)\mathbf{A}^{-T}\mathbf{r}\mathbf{r}^T\mathbf{A}^{-1} \tag{8}$$

$\mathbf{Q}$ describes a conic, and $\mathbf{A}^{-T}\mathbf{A}^{-1}$ is the image of the absolute conic [10, 8], from which the camera intrinsic parameters can be uniquely determined (see Appendix A).

Since

$$\begin{aligned} \mathbf{Q} &\cong \mathbf{A}^{-T}(\mathbf{I} - (r^2+1)\mathbf{r}\mathbf{r}^T)\mathbf{A}^{-1} \\ &= \mathbf{A}^{-T}\mathbf{r}_1\mathbf{r}_1^T + \mathbf{r}_2\mathbf{r}_2^T - r^2\mathbf{r}\mathbf{r}^T\mathbf{A}^{-1} \\ &= \mathbf{A}^{-T}\,[\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}]\,\mathrm{diag}(1, 1, -r^2)\,[\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}]^T\,\mathbf{A}^{-1} \, , \end{aligned}$$

$\mathbf{Q}$ has two positive and one negative eigenvalues.

Given $\mathbf{r}$, we can determine its intersection with the image plane. Let the intersection's normalized image coordinates be (x_c, y_c). We have

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} r_1/r_3 \\ r_2/r_3 \end{bmatrix} \tag{9}$$

where $\mathbf{r} = [r_1, r_2, r_3]^T$. Together with r, x_c and y_c determine the relative size and position of a ball up to the unknown depth.

r and (x_c, y_c) have 3 degrees of freedom. Since each conic provides 5 constraints, only 2 of them are for the intrinsic parameters. If we model the camera as having 5 intrinsic parameters, we need minimally 3 balls.

3 From Conics to the Absolute Conic: Optimal Estimation of Intrisic Parameters and Ball's Relative Size and Position

In the case of 3 balls, we have totally 14 independent unknown parameters, 5 intrinsic parameters and 9 parameters, $r(j), x_c(j), v_c(j), j = 1, 2, 3$, to model the relative sizes and positions of the 3 balls.

By the conventional wisdom, we model image noise as isotropic uniform Gaussian. The optimal estimation tries to find the unknown parameters so that the sum of the squared Euclidean distances between the image points and respective conics is minimized.

Three issues have to be considered. The first is that the Euclidean distance from a point to a conic is very complex to express explicitly. We will illustrate this later. The second issue is that since the distance is a non-linear function of the unknown parameters, we need to use nonlinear optimization algorithms which involves the differentials of the distance with respect to the unknown parameters. Here we use the Levenberg-Marquardt algorithm which only requires the computation of the first-order differentials [16]. The third issue is that non-linear iterative algorithms need good initial estimates.

Assuming that n points are extracted from the ball boundaries in the image, the cost function is defined as

$$C = \frac{1}{2}\sum_{i=1}^{n}\{(u_i - U_j(u_i, v_i))^2 + (v_i - V_j(u_i, v_i))^2\} \tag{10}$$

where u_i, v_i are the pixel coordinates of the i-th image point, and $U_j(u_i, v_i), V_j(u_i, v_i)$ are the coordinates of (u_i, v_i)'s closest point on the ellipse of the j-th ball, to which the i-th point belongs.

In its canonical form, an ellipse can be described by

$$\frac{U^2}{a^2} + \frac{V^2}{b^2} = 1$$

The distance from point (x, y) to the ellipse can be obtained by minimizing

$$(x - U)^2 + (y - V)^2 + \lambda(\frac{U^2}{a^2} + \frac{V^2}{b^2} - 1)$$

where λ is the Lagrange multiplier. Setting the differentials of the above expression with respect to U and V to be zeros yields a 4-th order polynomial equation about λ,

$$\frac{a^2x^2}{(a^2 + \lambda)^2} + \frac{b^2y^2}{(b^2 + \lambda)^2} = 1 \tag{11}$$

There are 4 solutions, 2 of which may be imaginary. We need to choose the solutions that provides the shortest real distance. Although analytic solutions for λ (and thus for U, V) are available, the expressions are long and complex. They become even more complex if we add the transform from a general ellipse to its canonical form and if we try to differentiate the cost function (10) with respect to the unknown parameters.

The strategy we adopt to resolve this difficulty is to approximate the differentials by numerical calculations. Using the current values of intrinsic parameters and the relative size and position of the balls, we can determine the transformation from the pixel image coordinate system to the canonical coordinate system of each ellipse. We can then further compute the closest point on the ellipse to a given point off the ellipse by solving (11) and find the closest real distance from the 4 solutions. Let the result be $U_j(u_i, v_i, p_1, ..., p_k, ..., p_{14})$, $V_j(u_i, v_i, p_1, ..., p_k, ..., p_{14})$. We then change one of the unknown parameters p_k by a small Δ, and compute the closest point on the ellipse again. Let the new coordinates be $U_j(u_i, v_i, p_1, ..., p_k + \Delta, ..., p_{14})$, $V_j(u_i, v_i, p_1, ..., p_k + \Delta, ..., p_{14})$. The differentials can be approximated by

$$\begin{aligned} \frac{dU_j(u_i, v_i)}{dp_k} &\approx (U_j(u_i, v_i, p_1, ..., p_k + \Delta, ..., p_{14}) \\ &- U_j(u_i, v_i, p_1, ..., p_k, ..., p_{14}))\frac{1}{\Delta}\ , \\ \frac{dV_j(u_i, v_i)}{dp_k} &\approx (V_j(u_i, v_i, p_1, ..., p_k + \Delta, ..., p_{14}) \\ &- V_j(u_i, v_i, p_1, ..., p_k, ..., p_{14}))\frac{1}{\Delta}\ . \end{aligned}$$

With the first-order differentials available, we can now use the Leverberg-Marquardt algorithm. The success of convergence to the global minimum depends on good initial estimates of the parameters. We found that if we assume that the image center is the principal point, pixels are square $(\alpha_u = \alpha_v)$ and the skew is zero $(b = 0)$, then the other parameters can be obtained by first recovering the ellipse and then solving a few equations. See Appendix B for details. Note that these assumptions are good approximations of the modern cameras, and empirically we found that they do not introduce errors that are so large that the global minimum cannot be reached.

Once the intrinsic parameters are known, we can determine the relative size and position of each ball. See Appendix C for details.

4 Implementation Details and Experimental Results

With all the equations given above and given in the appendices, we can now implement the algorithm. To be more numerically robust, we choose to normalize the image first by the initial estimate of the intrinsic matrix, then optimally estimate the parameters over the normalized image, and finally transform the obtained intrinsic parameters back to their original domains.

Suppose that the initial estimate of the intrinsic matrix is $\mathbf{A}_0$. We project all the image points to the normalized image by

$$\tilde{\mathbf{x}}_i = \mathbf{A}_0^{-1}\tilde{\mathbf{m}}_i,\ i = 1, ..., n.$$

The conic matrix $\mathbf{Q}(j)$ in (8) for the normalized image points of the j-th ball becomes

$$\mathbf{Q}'(j) \cong \mathbf{A}'^{-T}\mathbf{A}'^{-1} - (r(j)^2 + 1)\mathbf{A}'^{-T}\mathbf{r}(j)\mathbf{r}(j)^T\mathbf{A}'^{-1}.$$

where $\mathbf{A}'$ is the intrinsic matrix for the new image. The initial guess of $\mathbf{A}'$ used for the optimization is the identity matrix. From $\mathbf{Q}'(j)$, we can compute the initial estimate of $r(j)$ and $\mathbf{r}(j)$ by the formula in Appendix C. And $\mathbf{r}(j), j = 1, 2, 3$ is used to determine the initial estimates of $x_c(j), y_c(j), j = 1, 2, 3.$

Let the optimally estimated intrinsic matrix for the normalized image be $\mathbf{A}'$. The final intrinsic matrix is then

$$\mathbf{A} = \mathbf{A}_0\mathbf{A}' . \tag{12}$$

The implementation goes in the following steps: (1) extract boundary for each ball using, e.g., a “snake”, and sample (40-60) points for each boundary; (2) fit each point set by a conic by, e.g., Kanatani’s program downloaded from http://www.ail.cs.gunma-u.ac.jp/∼kanatani/; (3) choose the ball that is the farest from the image center and estimate the focal length using the formula in Appendix B; (4) normalize the image (points) by $\mathbf{A}_0$ with the focal length determined above; (5) estimate the relative size and position of each ball using the formula in Appendix C; (6) optimally estimate the 14 parameters in the normalized image; (7) project all the estimated parameters and their estimated variances back to the original digital image domain.

We have conducted experiments on both synthetic and real images. For the synthetic image (Fig. 2) of 1000×1000 pixels with a focal length of 1000 pixels, 3 balls are generated and isotropic uniform Gaussian noise of $\sigma = 2.0$ is added to the ball boundary points. Fig. 3 shows

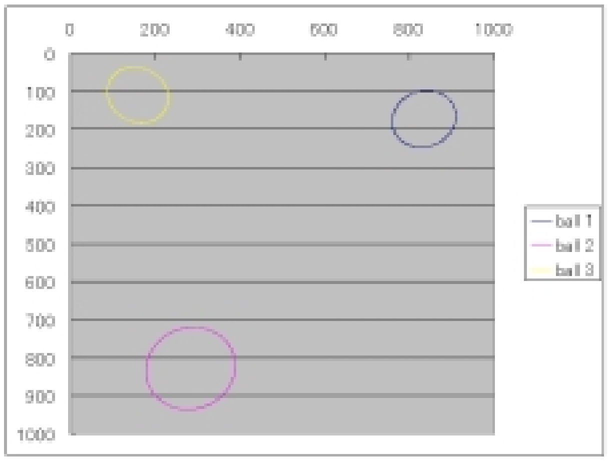

Figure 2: A simulation image of 3 balls. Gaussian noise of $\sigma = 2.0$ is added to the image points.

how α_u converges. The three tracks are respectively for the different initial estimates of focal length using the 3 different conics. The three tracks converge to the same minimum.

The results of optimization for Gaussian noise $\sigma = 2.0$, $\sigma = 1.0$ and $\sigma = 0.0$ are listed in Table 1. The initial and computed values are shown together with the variance estimated by the Levenberg-Marquartd algorithm. It can be seen from the table that the larger the Gaussian noise, the more the computed values deviate from the true values, and the larger the computed covariances.

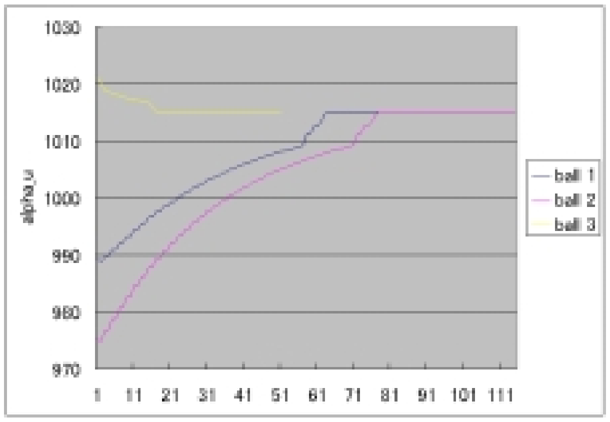

Figure 3: This figure shows how α_u converges. The three tracks are respectively for the different initial estimates of focal length using the 3 different conics. The three tracks converge to the same minimum.

Table 1

σ	param.	α_u	α_v	u_0	v_0	b
	true	1000	1000	500	500	0.0
0.0	init.	1010	1010	500	500	0.0
	final	1010	1012	502	501	-0.9
	var.	17.7	18.6	7.0	7.0	3.06
1.0	init.	995.7	995.7	500	500	0.0
	final	1014	1018	508	505	-6.9
	var.	17.9	18.8	7.1	7.1	3.13
2.0	init.	989.5	989.5	500	500	0.0
	final	1015	1020	511	507	-9.5
	var.	18.0	18.9	7.2	7.1	3.16

We also examined how the ball positions affect the final result. We moved the balls in Fig. 2 toward the image center in two steps and produced the images in Fig. 4. Fig. 5 shows how α_u starts from different initial values and converges to different values. It can be seen that the closer the balls are to the image center, the more the computed values deviate from the true values.

We have also done experiments on real images. One example is shown in Fig. 6. The camera is Power Shot Pro 70, a digital camera manufactured by Canon. The image size is 1536×1024 pixels. We used 60 points along the boundary for each ball. One of the balls is very close to the image center. It does not provide the initial estimate of

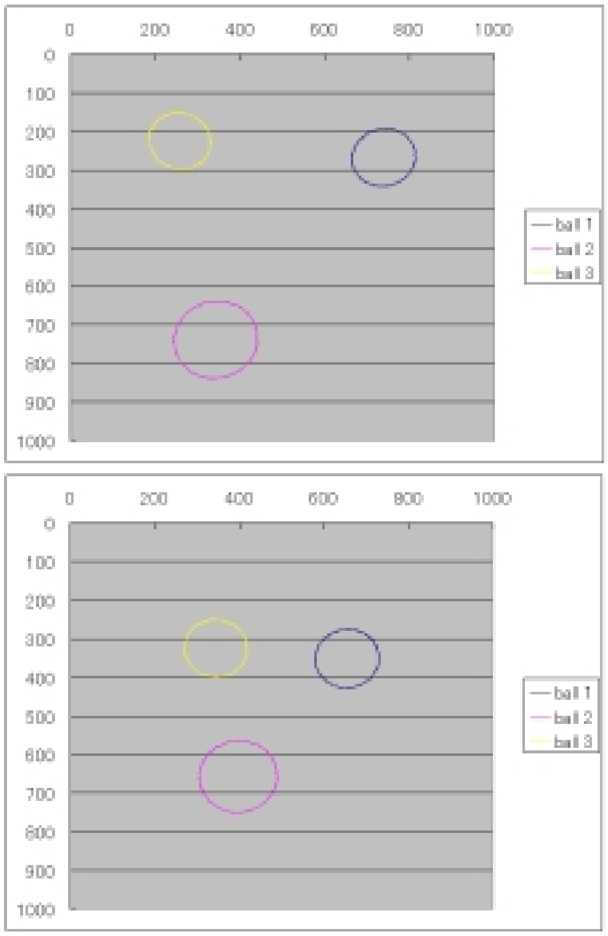

Figure 4: Two new simulation images with the balls closer to the image center.

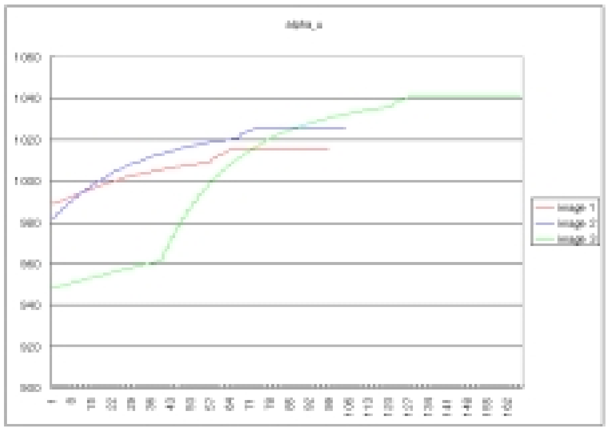

Figure 5: This figure shows the three tracks of convergence of α_u for image 1 (Fig. 2), image 2 and image 3 (Fig. 4). The closer the balls are to the image center, the more the computed value deviate from the true value.

focal length using the formula in Appendix B. This is natural because when the center coincides with the principal point, the ball's image is a circle, which does not provide necessary information to determine the focal length. This is why we choose the conic that is the farthest from the image center for providing the initial estimate of the intrinsic parameters.

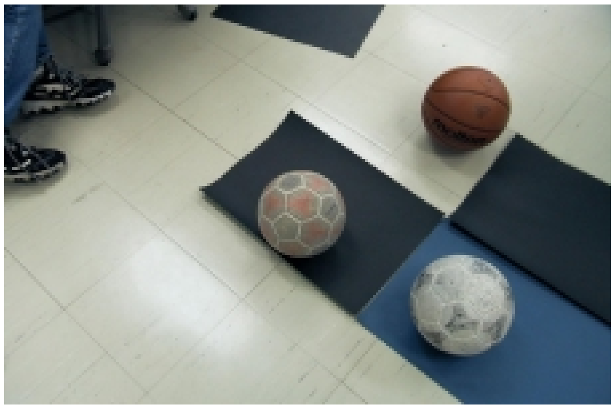

Figure 6: A real image of 3 balls. The image size is 1536×1024 pixels.

The optimally estimated values are listed in Table 2. The initial and computed values are shown together with the variance estimated by the Levenberg-Marquartd algorithm. Starting from the intial guess obtained from the other ball off the image center, all the values converged to the same minimum as in Table 2. The results look very promising. Experiments with other real images confirm that the algorithm is very robust.

Table 2

	α_u	α_v	u_0	v_0	skew
init.	1296	1296	768	512	0.0
final	1348	1346	767	514	0.19
var.	32.6	33.5	17.7	13.4	3.3

5 Conclusions

In this paper we have proposed a new flexible technique to calibrate camera's intrinsic parameters by taking a single image of 3 balls. A ball is projected onto the image as an ellipse, or a conic. It is known that the image of the absolute conic corresponds to the intrinsic matrix up to a scale factor. we proposed an algorithm to optimally determine the image of the absolute conic given the conics as the image of balls by minimizing the sum of squared Euclidean distances from the image points to the estimated ellipses. Experimental results with both synthetic and real images show that the algorithm is very effective. The advantages of this approach are that balls are ubiquitous and that there is no correspondence problem involved.

References

[1] P. Beardsley, D. Murray, and A. Zissermann. Camera calibration using multiple images. In Giulio Sandini, editor, *Proc. Second European Conf. Comput. Vision*, pages 312–320. Springer-Verlag, Lecture Notes in Computer Science 588, 1992.

[2] S. Bougnoux. From projective to euclidean space under any practical situation, a critism of self-calibration. In *Proc. Sixth Int'l Conf. Comput. Vision*, pages 790–796, Bombay, India, January 1998. Narosa Publishing House.

[3] N. Daucher, M Dhome, and J.T. Lapreste. Camera calibration from spheres images. In *Proc. Third European Conf. Comput. Vision*, pages 449–454, Stockholm, 1994.

[4] O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.

[5] R. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Trans. PAMI*, 19(2):133–135, February 1997.

[6] K. Kanatani. *Statistical Optimization for Geoometric Computation: Theory and Practice*. North-Holland, 1996.

[7] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 482–488, Santa Barbara, California, June 1998. IEEE.

[8] Q.-T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *Int'l J. Comput. Vision*, 22(3):261–289, 1997.

[9] S. D. Ma. A self-calibration technique for active vision systems. *IEEE Trans. RA*, 12:114–120, 1996.

[10] S.J. Maybank and O.D. Faugeras. A theory of self-calibration of a moving camera. *Int'l J. Comput. Vision*, 8(2):123–152, August 1992.

[11] M. Pennai. Camera calibration: A quick and easy way to determine the scale factor. *IEEE Trans. PAMI*, 13(12):1240–1245, 1991.

[12] M. Pollefeys, R. Koch, and L. van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. Sixth Int'l Conf. Comput. Vision*, pages 90–95, Bombay, India, January 1998. Narosa Publishing House.

[13] P Sturm and S Maybank. On plane-based camera calibration: A general algorithm, singularities and applications. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 432–437, Fort Collins, Colorado, USA, June 1999. IEEE.

[14] B. Trigg. Autocalibration from planar scenes. In *Proc. Fifth European Conf. Comput. Vision*, pages 89–105, Freiberg, Germany, June 1998.

[15] R. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 364–374, Miami, FL, June 1986. IEEE.

[16] W.H. Press B.P. Flannery S.A. Teukolsky W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[17] G.Q. Wei and S.D. Ma. A complete two-plane camera calibration method and experimental comparisons. In *Proc. Fourth Int'l Conf. Comput. Vision*, pages 439–446, Berlin, May 1993.

[18] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.

[19] Z Zhang. Flexible camera calibration by viewing planes from unknown orientations. In *Proc. Seventh Int'l Conf. Comput. Vision*, pages 666–673, Keryra, Greece, Oct. 1999.

A Computing Intrinsic Parameters from the Image of the Absolute Conic

The image of the absolute conic $\mathbf{A}^{-T}\mathbf{A}^{-1}$ can be expressed as

$$\mathbf{A}^{-T}\mathbf{A}^{-1} = \begin{bmatrix} \frac{1}{\alpha_u^2} & -\frac{b}{\alpha_u^2\alpha_v} & \frac{bv_0-u_0\alpha_v}{\alpha_u^2\alpha_v} \\ * & \frac{b^2}{\alpha_u^2\alpha_v^2}+\frac{1}{\alpha_v^2} & -\frac{b(bv_0-u_0\alpha_v)}{\alpha_u^2\alpha_v^2}-\frac{v_0}{\alpha_v^2} \\ * & * & \frac{(bv_0-u_0\alpha_v)^2}{\alpha_u^2\alpha_v^2}+\frac{v_0^2}{\alpha_v^2}+1 \end{bmatrix}$$

where the components indicated by * are omitted to save space. Note that the matrix is symmetric.

Define a symmetric matrix $\mathbf{C}$ with $C_{33}=1$ such that

$$\lambda\mathbf{C} = \mathbf{A}^{-T}\mathbf{A}^{-1}$$

From $\mathbf{C}$, it is straightforward to obtain

$$\begin{aligned} v_0 &= \frac{C_{12}C_{13}-C_{11}C_{23}}{C_{11}C_{22}-C_{12}^2} \\ \lambda &= \frac{C_{11}}{C_{11}-(C_{13}^2+v_0(C_{12}C_{13}-C_{11}C_{23}))} \\ \alpha_u &= \sqrt{\frac{1}{\lambda C_{11}}} \\ \alpha_v &= \sqrt{\frac{C_{11}}{\lambda(C_{11}C_{22}-C_{12}^2)}} \\ b &= -\lambda C_{12}\alpha_u^2\alpha_v \\ u_0 &= \frac{bv_0}{\alpha_u} - \lambda C_{13}\alpha_u^2 \end{aligned}$$

B Determining Focal Length Assuming Square Pixels, Zero Skew and Known Principal Point

When $\alpha_u=\alpha_v=\alpha$ and $b=0$, $\mathbf{A}^{-T}\mathbf{A}^{-1}$ becomes

$$\mathbf{A}^{-T}\mathbf{A}^{-1} = \begin{bmatrix} \frac{1}{\alpha^2} & 0 & -\frac{u_0}{\alpha^2} \\ 0 & \frac{1}{\alpha^2} & -\frac{v_0}{\alpha^2} \\ -\frac{u_0}{\alpha^2} & -\frac{v_0}{\alpha^2} & \frac{u_0^2}{\alpha^2}+\frac{v_0^2}{\alpha^2}+1 \end{bmatrix}$$

From (8), we have

$$k\mathbf{Q} = \mathbf{A}^{-T}\mathbf{A}^{-1} - \mathbf{a}\mathbf{a}^T$$

where $\mathbf{a}=[a_1,a_2,a_3]^T=\sqrt{r^2+1}\mathbf{A}^{-T}\mathbf{r}$. Using the upper left 2×2 submatrix, we obtain

$$\begin{aligned} a_1^2 &= \frac{1}{\alpha^2} - kQ_{11} \\ a_2^2 &= \frac{1}{\alpha^2} - kQ_{22} \\ a_1a_2 &= -kQ_{12} \end{aligned}$$

which lead to

$$\alpha^2 k = \frac{Q_{11}+Q_{22}\pm\sqrt{(Q_{11}-Q_{22})^2+4Q_{12}^2}}{2(Q_{11}Q_{22}-Q_{12}^2)}$$

Experiments have shown that $\alpha^2 k = \frac{Q_{11}+Q_{22}-\sqrt{(Q_{11}-Q_{22})^2+4Q_{12}^2}}{2(Q_{11}Q_{22}-Q_{12}^2)}$ is always the right choice. Note that when the ball is close to the image center, the image becomes close to a circle and $Q_{11}Q_{22}-Q_{12}^2$ becomes close to zero. It means that the focal length cannot be obtained in this case by solving the above equation.

Using the rightmost column vector, we obtain

$$\begin{aligned} -a_1a_3 &= kQ_{13} + \frac{u_0}{\alpha^2} \\ -a_1a_3 &= kQ_{23} + \frac{v_0}{\alpha^2} \\ -a_3^2 &= kQ_{33} - 1 - \frac{u_0^2}{\alpha^2} - \frac{v_0^2}{\alpha^2} \end{aligned}$$

which lead to

$$\begin{aligned}\alpha^2 &= \alpha^2 k \frac{-Q_{13}Q_{23} + Q_{12}Q_{33}}{Q_{12}} - \frac{Q_{13}v_0 + Q_{23}u_0}{Q_{12}} \\ &- u_0^2 - v_0^2 - \frac{u_0 v_0}{\alpha^2 k Q_{12}}\end{aligned}$$

With $\alpha^2 k$ already expressed in terms of $\mathbf{Q}$'s components, $\alpha(\alpha > 0)$ can be easily obtained.

C Determining Ball's Relative Size and Position

In this section, we show how to determine the relative size and position of a ball, given its conic equation and the intrinsic matrix.

From (8), we have

$$k\mathbf{Q} = \mathbf{A}^{-T}\mathbf{A}^{-1} - (r^2+1)\mathbf{A}^{-T}\mathbf{r}\mathbf{r}^T\mathbf{A}^{-1}$$

Since $\mathbf{A}$ is known, we have

$$k\mathbf{A}^T\mathbf{Q}\mathbf{A} = \mathbf{I} - (r^2+1)\mathbf{r}\mathbf{r}^T$$

The left side can be decomposed as

$$\mathbf{U}\text{diag}(k\lambda_1, k\lambda_2, k\lambda_3)\mathbf{U}^T$$

and the right side can be written as

$$[\,\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}\,]\,\text{diag}(1, 1, -r^2)\,[\,\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}\,]^T$$

It is easy to see that

$$\mathbf{U} = [\,\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}\,]$$

and

$$r = \sqrt{-\frac{\lambda_3}{\lambda_1}}$$

D Fitting Conics to Ball Boundaries in Image

A conic is described by

$$\begin{aligned}Q(u,v) &= Au^2 + 2Buv + Cv^2 + 2Du + 2Ev + F \\ &= \tilde{\mathbf{u}}^T\mathbf{Q}\tilde{\mathbf{u}} = 0\end{aligned}$$

where

$$\mathbf{Q} = \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix},$$

and

$$\tilde{\mathbf{u}} = [u, v, 1]^T .$$

In particular, for an ellipse, $B^2 - AC < 0$.

The simplest way to fit a conic to the point set $\{\mathbf{u}_i\} = \{(u_i, v_i)\}(i = 1, ..., m)$ is to minimize

$$E = \sum_{i=1}^{m} Q^2(u_i, v_i) = \sum_{i=1}^{m} \mathbf{q}^T\mathbf{p}_i\mathbf{p}_i^T\mathbf{q}$$

where

$$\begin{aligned}\mathbf{q} &= [A, B, C, D, E, F]^T, \\ \mathbf{p}_i &= [u_i^2, 2u_iv_i, v_i^2, 2u_i, 2v_i, 1]^T\end{aligned}$$

This problem has a closed-form solution. The optimal $\mathbf{q}$ is the eigenvector associated with smallest eigenvalue of matrix $\sum_{i=1}^{m}\mathbf{p}_i\mathbf{p}_i^T$. Note that the scale of $\mathbf{q}$ is arbitrary.

While the algorithm is simple, the result is not guaranteed to be optimal as E does not correspond to a geometrically meaningful quantity. Geometrically speaking, the optimal fitting is one that the conic goes as close as possible to the image points. Thus we should minimize the squared Euclidean distances from the image points to the ellipse [18]. Kanatani proves that minimizing the squared Euclidean distances is equivalent to estimating the conic by the renormalization approach assuming the isotropic uniform Gaussian distribution of noise in image position [6].