Camera calibration using spheres: A semi-definite programming approach

Motilal Agrawal[*]
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
agrawal@ai.sri.com

Larry S. Davis
Univ. of Maryland
Dept. of Computer Science
College Park, MD 20742
lsd@cs.umd.edu

Abstract

Vision algorithms utilizing camera networks with a common field of view are becoming increasingly feasible and important. Calibration of such camera networks is a challenging and cumbersome task. The current approaches for calibration using planes or a known 3D target may not be feasible as these objects may not be simultaneously visible in all the cameras. In this paper, we present a new algorithm to calibrate cameras using occluding contours of spheres. In general, an occluding contour of a sphere projects to an ellipse in the image. Our algorithm uses the projection of the occluding contours of three spheres and solves for the intrinsic parameters and the locations of the spheres. The problem is formulated in the dual space and the parameters are solved for optimally and efficiently using semi-definite programming. The technique is flexible, accurate and easy to use. In addition, since the contour of a sphere is simultaneously visible in all the cameras, our approach can greatly simplify calibration of multiple cameras with a common field of view. Experimental results from computer simulated data and real world data, both for a single camera and multiple cameras, are presented.

1 Introduction

The recovery of $3D$ information from $2D$ images is a fundamental problem in computer vision. This recovery of metric information requires knowledge of the camera parameters – also known as camera calibration. A real world camera can be approximated by a pinhole camera model. This model projects points in $3D$ by a perspective projection. The parameters of the camera to be recovered are classified as *intrinsic* and *extrinsic* parameters. The extrinsic parameters relate the world coordinates to the camera orientation and position and the intrinsic parameters define the imaging geometry of the camera. In the pinhole camera model, the intrinsic parameters of the camera are modelled by five terms. (Most non-linearities in cameras can be modelled by an additional radial distortion term.) There has been much work on camera calibration, both in photogrammetry and computer vision. These work can be classified into two categories.

In the first category are methods which use a calibration object. These can be further classified into two subcategories: methods which use a calibration object with a fixed $3D$ geometry [14, 21] and methods with "generic" 3D geometry, eg. coplanar points. Recently, for example, Zhang [23] has shown that it is possible to calibrate a camera using a planar point pattern shown at a few different orientations.

In the second category are methods which do not use a calibration object and are generally referred to as self-calibration [12, 16]. In this approach a camera is moved in a static scene. If the internal parameters of the camera do not change, then the rigidity of the scene provides two constraints on the internal parameters through Kruppa's equations. Correspondences between three images are then sufficient to recover the camera parameters. In some cases, even if some of the camera parameters do change during the motion, they can still be recovered [18]. However, since a large number of parameters need to be estimated, this method is very susceptible to noise and is often unstable.

In this paper, we use spheres as calibration object. The primary motivation for using spheres as calibration objects stems from the need for simultaneously calibrating multiple cameras mounted far apart from one another. Over the last few years, cameras have become ubiquitous due to their declining costs, so that vision systems using multiple cameras are becoming cost effective. Example of multi-view vision systems include the $3D$ room developed by Kanade et al [19] and the Keck Laboratory (Davis et al [9]).

Calibration of such a large number of cameras using a $3D$ reference object is tedious and cumbersome. In addi-

[*] This work was carried out while the author was a graduate student at the University of Maryland

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

tion, it is very difficult to calibrate all the cameras simultaneously using these reference objects, as all the points on the calibration object are not visible simultaneously in all views. Similarly, planar patterns [23] cannot be used for calibrating these multi-perspective vision systems, as a plane is visible to only those cameras that it lies in front of. Points are a natural choice since a point would be visible to all the cameras simultaneously [2]. But this technique faces various problems. It is very difficult to localize the images of the points and, since, Euclidean constraints are not available, these point correspondences provide only weak calibration (intrinsic parameters cannot be recovered). In addition, this technique usually needs a large number of images of these points. Recently, Zhang [24] has presented a method for calibration using $1D$ objects which can potentially be used for simultaneously calibrating a large number of cameras. However, this requires that one point of the line be fixed, which may be difficult in practice.

The occluding contours of spheres, in addition to providing all the advantages of points, can also be localized accurately. The image of an occluding contour of a sphere is an ellipse in general, and is closely related to the absolute conic which determines the intrinsic parameters of the camera. In addition, as long as the sphere is placed in the common field of view of the cameras, its occluding contours are visible from any position. Hence they can be used to accurately calibrate multiple cameras mounted at arbitrary locations simultaneously. We present an algorithm that uses three images of the sphere to simultaneously calibrate the intrinsic and the extrinsic parameters of the camera. Our approach is based on semi-definite programming and solves for the parameters of the camera directly without making any assumptions about the parameters to be estimated. This is achieved by formulating the problem in dual space.

Spheres have been used previously for calibration. [8] and [3] have used four parameters to model the intrinsic parameters of the camera, which are calculated in a multi-step approach. First, the aspect ratio of the camera is estimated. Then, using the fact that the major axes of the ellipse intersect at the principal point, the principal point is estimated, and finally, the focal length is calculated. The more recent work of Teramoto and Xu [20] is closest to our work. However, in their approach the minimization is accomplished using general-purpose non-linear minimization and needs a good initial estimate to start the minimization. They obtain the initial estimate by assuming that the image center is the principal point, the skew is zero, and the pixels are square. While this is a reasonable assumption, this may still cause the optimization procedure to get stuck at a local minimum. In addition, the principal point is known to vary with the camera zoom. In a recent paper, Hartley and Kaucic [13] have pointed out the sensitivity of calibration to principal point estimation. They show that small changes in the principal point position can cause very large changes in the estimated focal length. Our dual space formulation, on the other hand, is general enough for any camera which can be modelled by five intrinsic parameters and does not need an initial estimate. We also illustrate how to recover the extrinsic parameters in a multi-camera setup.

The remainder of the paper is organized as follows. In the next section, we present the preliminaries of the pinhole camera model and quadrics. In Section 3, we derive the basic equations governing the projection of the occluding contour of the sphere. Sections 4 and 6 present the algorithm for solving these equations. The algorithm uses semi-definite programming for minimization, a brief introduction to which is presented in Section 5. Experimental results for real and simulated data are presented in Section 7. Finally, Section 8 concludes the paper with suggestions for future work.

2 Preliminaries

2.1 Pinhole Camera Model

Let X be a world point in homogenous coordinates and x be its projection in the camera with projection matrix P; this can be represented as

$$x = PX \tag{1}$$

The camera matrix P can be decomposed as

$$\begin{aligned} P &= AR\begin{bmatrix} I & -t \end{bmatrix} && (2)\\ A &= \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} && (3)\\ R^tR &= I && (4) \end{aligned}$$

The upper triangular matrix A is the matrix of intrinsic parameters and R and t are the rotation and translation of the camera respectively. Note that we are ignoring non-linear distortion in our model.

2.2 Quadrics

In homogenous coordinates a quadric is represented by a 4×4 symmetric matric Q; the equation of the quadric is

$$X^tQX = 0 \tag{5}$$

The dual of the quadric Q is another quadric Q^*. Dual quadrics are equations on planes: the tangent planes π to the point quadric Q satisfy $\pi^tQ^*\pi = 0$, where Q^* =adjoint Q, or Q^{-1} if Q is invertible.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

In particular, a sphere is a quadric with $S \equiv Q$ matrix of the form

$$S \equiv Q = \begin{bmatrix} I & -a \\ -a^t & \gamma \end{bmatrix} \tag{6}$$

This sphere S has center at a and the square of its radius is $a^t a - \gamma$. Note that for real spheres we must have $a^t a - \gamma > 0$. The dual of this sphere is the quadric S^* where

$$S^* = S^{-1} = \begin{bmatrix} I - \beta a a^t & -\beta a \\ -\beta a^t & -\beta \end{bmatrix} \quad \text{where} \quad \beta = \frac{1}{a^t a - \gamma} \tag{7}$$

Note that $\beta > 0$ for all real spheres. Under the camera matrix P, the apparent contour (also called outline) of a quadric is a conic C with dual C^* given by

$$C^* = PQ^*P^t \tag{8}$$

The proof can be found in [14].

2.3 Absolute Dual Quadric

The absolute quadric is an imaginary quadric (i.e. with no real points) and is defined by its dual representation

$$Q^*_\infty = \begin{bmatrix} I & 0 \\ 0^t & 0 \end{bmatrix} \tag{9}$$

The dual image of this absolute quadric (DIAC) is a conic represented by ω^*. This DIAC plays an important role in camera calibration. It can be shown [14] that under the camera matrix $P = AR\begin{bmatrix} I & -t \end{bmatrix}$, the DIAC is given by

$$\omega^* = AA^t \tag{10}$$

Hence, if ω^* is known, the intrinsic parameters A are easily obtained by (uniquely) finding the Cholesky factorization of ω^*.

3 Calibration using the apparent contour of a sphere

The apparent contour of a sphere with center at a under the camera matrix $P = AR\begin{bmatrix} I & -t \end{bmatrix}$is obtained by substituting the equation for the dual sphere (equation 7) into equation 8:

$$C^* = P \begin{bmatrix} I - \beta a a^t & -\beta a \\ -\beta a^t & -\beta \end{bmatrix} P^t$$

Substituting for P from equation 2 gives

$$\begin{aligned} C^* &= AR\begin{bmatrix} I & -t \end{bmatrix} \begin{bmatrix} I - \beta a a^t & -\beta a \\ -\beta a^t & -\beta \end{bmatrix} \begin{bmatrix} I \\ -t^t \end{bmatrix} R^t A^t \\ &= AR\begin{bmatrix} I & -t \end{bmatrix} \begin{bmatrix} I - \beta a a^t + \beta a t^t \\ -\beta a^t + \beta t^t \end{bmatrix} R^t A^t \\ &= AR\left[I - \beta\left(a - t\right)\left(a^t - t^t\right)\right] R^t A^t \end{aligned} \tag{11}$$

Let $c = a - t$. c is then the relative position of the sphere center with respect to the camera center. Substituting c and using the fact that R is orthogonal (equation 4), equation 11 becomes

$$\begin{aligned} C^* &= A\left[I - \beta R c c^t R^t\right] A^t \\ &= AA^t - \left(\sqrt{\beta} ARc\right)\left(\sqrt{\beta} ARc\right)^t \end{aligned} \tag{12}$$

Now, let v be the vector $\sqrt{\beta} ARc$. Therefore, the above equation may be written as

$$v = \sqrt{\beta} ARc \tag{13}$$

$$C^* = AA^t - vv^t \tag{14}$$

Equation 14 is the basis for our calibration approach and several key observations can be made about this equation.

Since the quadric C^* is defined up to an arbitrary scale, the equality in equation 14 is only up to an arbitrary scale factor. The left-hand-side (lhs) and right-hand-side (rhs) of this equation are all symmetric matrices. The lhs is the dual of the occluding contour of the sphere and can be calculated by fitting an ellipse in the image and taking its inverse as the dual. The rank of C^* is 3. The rhs is the sum of two matrices. The first, being the DIAC, is of rank 3 and the second term, being the outer product (vv^t), is of rank 1. Suppose the internal parameters of the camera are fixed while the camera images the sphere at different locations. Then the first term (AA^t) remains the same while the vector v, which depends on the translation and rotation of the sphere with respect to the camera, varies. Hence if we image the sphere at multiple positions, it is possible to recover the internal parameters.

We need a minimum of three such images in order to perform this calibration; this can be seen by a counting argument. Since the internal parameters of the camera are constant, AA^t is fixed. Therefore, there are six unknowns for AA^t. (Note that matrix A is also defined up to a scale). Each image of the sphere results in six constraints. (since all the matrices in equation 14 are symmetric). In addition each equation adds three unknowns for the v vector and one unknown for the global scale. Therefore, the number of unknowns for N such images (corresponding to N equations) is $6 + 4N$ and the number of constraints is $6N$. Hence, in order to solve this system of equations, we must have

$$\begin{aligned} 6N &\geq 6 + 4N \\ N &\geq 3 \end{aligned}$$

The above proof is only an argument by counting and does not give a procedure for finding AA^t and the extrinsic parameters, which is presented next.

4 Closed form solution

Suppose a sphere is viewed at three different positions by a camera without changing its internal parameters. We proceed by extracting the occluding contour of the spheres in the images. This can be accomplished by finding the edges in the image and fitting an ellipse to the edge points belonging to the ellipse. Fitting an ellipse to points is a classical problem and an accurate fit may be obtained using the method described in [10, 15]. As a result of the ellipse fitting, we obtain the quadrics C_1, C_2, C_3. The inverses of these quadrics C_1^*, C_2^*, C_3^* then give the dual quadric. Given these dual quadrics, we can solve for the intrinsic and extrinsic parameters. Let the three corresponding equations be

$$\begin{aligned} \kappa_1 C_1^* &= AA^t - v_1 v_1^t \\ \kappa_2 C_2^* &= AA^t - v_2 v_2^t \\ \kappa_3 C_3^* &= AA^t - v_3 v_3^t \end{aligned} \tag{15}$$

where $\kappa_{1,2,3}$ are the unknown scale factors.

4.1 Solving for the scale factors

First, we solve for the scale factors κ_1, κ_2, κ_3. We observe that

$$C_{ij} \equiv \kappa_i C_i^* - \kappa_j C_j^* \quad = \quad v_j v_j^t - v_i v_i^t \tag{16}$$

Hence, the rank of C_{ij} is at most 2. Therefore κ_j/κ_i is a generalized eigenvalue [11] of C_i^* and C_j^*. A 3×3 matrix has, in general, three generalized eigenvalues. Hence, there will be three values for κ_2/κ_1 and κ_3/κ_1. We arbitrarily fix $\kappa_1 = 1$ and choose values for κ_2 and κ_3 such that it is consistent with the generalized eigenvalue for C_2^* and C_3^*.

4.2 Solving for the v's

From equation 14, it is easy to see that $v = \sqrt{\beta} AR(a-t)$ is the projection of the center of the sphere in the image in homogenous coordinates. Note that, in general, the center of the sphere does not project to the center of the ellipse. Here we will illustrate how to obtain the projection of the center for each of the three spheres.

The symmetric matrix C_{ij} can be factorized by SVD as $C_{ij} = U_{ij} S_{ij} U_{ij}^t$ where S_{ij} is a diagonal matrix of the singular values and $U_{ij} = \left[\begin{array}{ccc} e_{ij}^1 & e_{ij}^2 & e_{ij}^3 \end{array} \right]$ is orthonormal. According to equation 16, for any $i \neq j$ the rank of C_{ij} is 2. Therefore $S_{ij}(3,3) = 0$ and

$$S_{ij} \quad = \quad U_{ij}^t \left[v_j v_j^t - v_i v_i^t \right] U_{ij}$$

Let $p_j = U_{ij}^t v_j$ and $p_i = U_{ij}^t v_i$. Then

$$\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & 0 \end{bmatrix} \quad = \quad p_j p_j^t - p_i p_i^t$$

Since $p_j \neq p_i$, from the above equation it is easy to see that $p_j^t = \left[\begin{array}{ccc} \alpha_j & \beta_j & 0 \end{array} \right]$ and $p_i^t = \left[\begin{array}{ccc} \alpha_i & \beta_i & 0 \end{array} \right]$ Since $v_j = U_{ij} p_j$, therefore v_j is a linear combination of the first two eigenvectors of C_{ij}. The same also applies to v_i:

$$v_i = \gamma_1 e_{ij}^1 + \gamma_2 e_{ij}^2$$

For an additional view k of the sphere, we obtain C_{ik} and therefore v_i is also a linear combination of the first two eigenvectors of C_{ik}:

$$v_i = \delta_1 e_{ik}^1 + \delta_2 e_{ik}^2$$

From these two equations, we can solve for v_i up to a scale factor $v_i = \alpha_i \hat{v}_i$; this is the projection of the center of the sphere onto the image.

4.3 Solving for the internal parameters

From the previous section $v_i = \alpha_i \hat{v}_i$ where α_i is the scale parameter which needs to be computed. Let $V_i = \hat{v}_i \hat{v}_i^t$. Then

$$\begin{aligned} v_i v_i^t &= \alpha_i^2 \hat{v}_i \hat{v}_i^t \\ &= \alpha_i^2 V_i \end{aligned}$$

Therefore equations 15 become

$$\begin{aligned} C_1^* &= AA^t - \sigma_1 V_1 \quad && \sigma_1 = \alpha_1^2 > 0 \\ \kappa_2 C_2^* &= AA^t - \sigma_2 V_2 \quad && \sigma_2 = \alpha_2^2 > 0 \\ \kappa_3 C_3^* &= AA^t - \sigma_3 V_3 \quad && \sigma_3 = \alpha_3^2 > 0 \end{aligned} \tag{17}$$

In solving for A from these equations, we must take note of the fact that AA^t is symmetric positive definite and $\sigma_i > 0$. Therefore, our solution must obey these constraints. These constraints fit naturally into the formulation of semi-definite programming, a brief introduction to which is provided below.

5 Semi-definite programming

Semi-definite programming (denoted SDP) is an extension of linear programming (LP) where the nonnegativity constraints are replaced by positive semi-definiteness constraints on matrix variables. SDP is a very powerful tool that has found applications in positive definite completion problems, maximum entropy estimation and bounds for hard combinatorial problems; see e.g. the survey of Vandenberghe and Boyd [22].

The standard dual form of SDP can be expressed as minimizing a linear function of a variable $x \in \mathcal{R}^m$ subject to a matrix inequality

$$\begin{array}{ll} \text{minimize} & c^t x \\ \text{subject to} & F(x) \succeq 0 \\ F(x) = F_0 + & \sum_{i=1}^m x_i F_i \end{array}$$

The problem data are the vector $c \in \mathcal{R}^m$ and $m+1$ symmetric matrices $F_0, \ldots, F_m \in \mathcal{R}^{n \times n}$. The inequality sign in $F(x) \succeq 0$ means that $F(x)$ is symmetric positive definite.

Unlike LP, SDP is a nonlinear convex programming problem, because the feasible boundary (the cone of positive semi-definite matrices) is nonlinear. Nonetheless, SDP shares a key feature with LP – SDP can be effectively solved by generalizing interior-point methods developed originally for LP. The credit for this important discovery goes primarily to Nesterov and Nemirovski [17] and Alizadeh [1]. The fact that SDP can be efficiently solved by interior–point methods, both in practice and, from a theoretical point of view, *exactly* in polynomial time, has driven the recent surge of interest in the subject.

5.1 Norm minimization using SDP

Suppose a matrix $A(x)$ depends affinely on $x \in \mathcal{R}^k$: $A(x) = A_0 + x_1 A_1 + \cdots + x_k A_k \in \mathcal{R}^{p \times q}$, and we wish to minimize the spectral norm (maximal singular value) $\|A(x)\|$. This can be cast as the semi-definite program

$$\begin{array}{ll} \text{minimize} & \zeta \\ \text{subject to} & \begin{bmatrix} \zeta I & A(x) \\ A(x)^t & \zeta I \end{bmatrix} \succeq 0 \end{array}$$

If $B(x)$ is another matrix which depends linearly on x and we want to minimize $\|A(x)\| + \|B(x)\|$, then the corresponding semi-definite program to be solved is

$$\begin{array}{ll} \text{minimize} & \zeta + \eta \\ \text{subject to} & \begin{bmatrix} \zeta I & A(x) & & \\ A(x)^t & \zeta I & & \\ & & \eta I & B(x) \\ & & B(x)^t & \eta I \end{bmatrix} \succeq 0 \end{array}$$

6 Complete calibration using semi-definite programming

We solve for the scale parameters $\sigma_{1,2,3}$ and refine $\kappa_{2,3}$ using SDP-based norm minimization. From equation 17

$$C_1^* + \sigma_1 V_1 = \kappa_2 C_2^* + \sigma_2 V_2 = \kappa_3 C_3^* + \sigma_3 V_3 = AA^t$$

Therefore we want to solve for $\sigma_{1,2,3}$, $\kappa_{2,3}$ for the SDP problem:

$$\begin{array}{l} \text{minimize } \|A\| + \|B\| + \|C\| \\ \text{where} \\ A = C_1^* + \sigma_1 V_1 - \kappa_2 C_2^* - \sigma_2 V_2 \\ B = C_1^* + \sigma_1 V_1 - \kappa_3 C_3^* - \sigma_3 V_3 \\ C = \kappa_2 C_2^* + \sigma_2 V_2 - \kappa_3 C_3^* - \sigma_3 V_3 \end{array} \tag{18}$$

subject to the constraints

$$\begin{array}{lcl} C_1^* + \sigma_1 V_1 & \succeq & 0 \\ \kappa_2 C_2^* + \sigma_2 V_2 & \succeq & 0 \\ \kappa_3 C_3^* + \sigma_3 V_3 & \succeq & 0 \\ \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3) & \succeq & 0 \end{array} \tag{19}$$

The corresponding dual SDP can be written in standard form and is omitted here due to space restrictions. This SDP is minimized to obtain the refined scale parameters. AA^t can then be obtained from these equations in a least squares sense, which can then be used to obtain A using Cholesky decomposition.

$$AA^t = \frac{1}{3}\left(C_1^* + \sigma_1 V_1 + \kappa_2 C_2^* + \sigma_2 V_2 + \kappa_3 C_3^* + \sigma_3 V_3\right)$$

6.1 Calibration with known principal point

Most real world cameras have zero skew ($s = 0$). In addition, suppose the principal point (x_0, y_0) is also known. Then we need to obtain the focal length of the camera in the x and y direction $(\alpha_x \alpha_y)$. The DIAC for this restricted case can be written as

$$AA^t = \begin{bmatrix} \alpha_x^2 + x_0^2 & x_0 y_0 & x_0 \\ x_0 y_0 & \alpha_y^2 + y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix}$$

Therefore AA^t can be decomposed as a linear combination of three matrices.

$$\begin{array}{rcl} AA^t & = & \alpha_x^2 F_x + \alpha_y^2 F_y + F_0 \\ F_x & = & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ F_y & = & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ F_0 & = & \begin{bmatrix} x_0^2 & x_0 y_0 & x_0 \\ x_0 y_0 & y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix} \end{array}$$

Since AA^t can be written as a linear combination, the focal lengths and the scale parameters can be computed by solving the following norm minimization problem:

$$\begin{array}{l} \text{minimize } \|P\| + \|Q\| + \|R\| \\ \text{where} \\ P = \kappa_1 C_1^* + \sigma_1 V_1 - \left(\alpha_x^2 F_x + \alpha_y^2 F_y + F_0\right) \\ Q = \kappa_2 C_2^* + \sigma_2 V_2 - \left(\alpha_x^2 F_x + \alpha_y^2 F_y + F_0\right) \\ R = \kappa_3 C_3^* + \sigma_3 V_3 - \left(\alpha_x^2 F_x + \alpha_y^2 F_y + F_0\right) \end{array} \tag{20}$$

subject to the constraints

$$\operatorname{diag}\left(\sigma_1, \sigma_2, \sigma_3, \alpha_x^2, \alpha_y^2\right) \succeq 0 \tag{21}$$

6.2 Multi-camera calibration

Our algorithm can be used to calibrate multiple cameras simultaneously. This is achieved by imaging the sphere at three or more locations in the common field of view of the cameras. Thus, we can obtain the internal camera parameters A and the vectors v for each camera. Since the same sphere is viewed by each of the cameras, we can set the radius of the sphere to unity, i.e. $\beta = 1$. (Thus we have metric calibration up to scale). Then the $3D$ location of the center of the sphere with respect to the camera reference frame can be obtained by setting $t = 0$ and $R = I$ in equation 13.

$$a_i \quad = \quad A^{-1} v_i$$

These $3D$ point sets can then be registered to obtain the rotation and translation parameters for the cameras. Registering $3D$ point sets is a common problem in computer vision. The case of two point sets has been analytically solved by several authors [5]. This pairwise registration algorithm can be sequentially applied pair by pair for registering multiple point sets. However, this usually leads to propagation and cumulation of registration errors. This limitation can be overcome by using a global multiple point set registration algorithm [7, 4].

7 Experimental results

The proposed algorithm has been implemented and tested on both computer simulated and real data. We have used CSDP [6], a freely available implementation for solving semi-definite programming problems.

7.1 Simulated data

For the computer simulation, the intrinsic parameters of the camera are shown in the first row of Table 1. A sphere was then ray-traced at three different locations to obtain three images of the spheres. The image resolution was 640×480. Figure 1 shows the ray traced images of the three spheres. The contours of the ellipses were then extracted using Canny's edge detector and ellipses were fitted to these contours using a least squares ellipse fitting algorithm. The camera parameters obtained by our algorithm are shown in the second row of the table. The third row displays the relative error in the parameters obtained with respect to the focal length(which is taken to be 1000) (The relative difference with respect to the focal length rather than the absolute error is a meaningful error measure as argued by Zhang [24]). The relative error in this case is less than 1%. The error is primarily due to the fact that our extracted edges are correct to within a pixel. As we are looking at occluding contours, the external zero of the laplacian will probably give us a better localization. Sub-pixel edge detection methods will also result in further reduction of the errors.

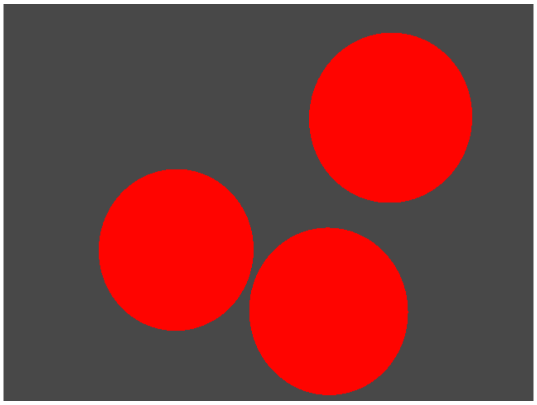


Figure 1. Ray traced images of three spheres

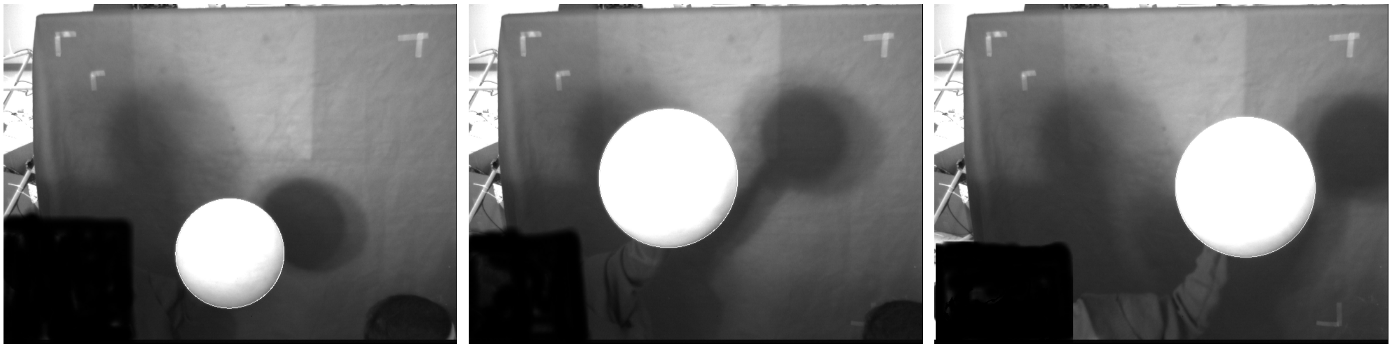
Table 1. Experimental results for synthetic data

Data	α_x	α_y	s	x_0	y_0
Ground truth	1000	1050	0.1	320	240
Calculated	992	1041	0.3	321	238
Relative difference	0.8%	0.9%	0.02%	0.01%	0.02%

7.2 Real Data

For the experiments with real data, we obtained a $12''$ diameter globe lamp. This globe lamp was then shown to the camera at three different locations. The three images are displayed in Figure 2. Edges were detected to obtain the contour of the ellipse and the parameters of the ellipse were obtained using ellipse fitting. The calculated intrinsic parameters using these images are listed in the first row of Table 2. Since ground truth is not available, in order to validate our parameters we obtained one more set of images. The extracted edges of the contours for the second set is shown in Figure 3. The parameters calculated using these three ellipses are given in the second row of Table 2 and the third row shows the relative difference in the parameters obtained using both sets (the focal length was taken to be 1151). The relative difference, in this case, is less than 2%.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE



(a) First Image (b) Second Image (c) Third Image

Figure 2. Images used for calibration

The difference in these two results can be attributed to image noise, non-linear distortion and edge localization error.

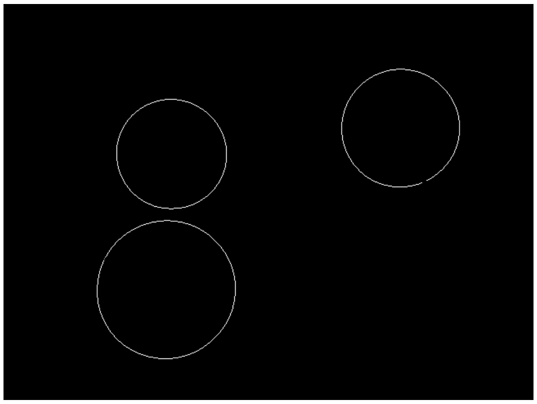


Figure 3. Second set of images

Table 2. Intrinsic parameters for real data

Set	α_x	α_y	s	x_0	y_0
Set1	1151.0	1151.9	3.03	373.7	222.5
Set2	1131.0	1133.0	2.26	363.1	228.7
Relative difference	1.73%	1.64%	0.07%	0.92%	0.54%

7.3 Multiple Camera Calibration

As pointed out earlier, our algorithm can be used to calibrate multiple cameras simultaneously. We have tested our multiple camera calibration on a pair of cameras in a stereo configuration. The cameras were placed vertically about one foot apart. We used six images of the sphere. We need only three spheres for calibration, but increasing the number of spheres results in increased accuracy of registration. The pairwise $3D$ registration algorithm [5] was used to calculate the rotational and translational parameters of the second camera with respect to the first using the six centers of the spheres. The residual error of registration is defined as the average distance between the first point set and the second point set after registration. This error for our case was about 0.5 inch.

In addition, we selected 20 random points in the background of the first image and hand matched them to find the corresponding points in the second image. The distance between the corresponding point and the epipolar line is a measure of the goodness of the calibration parameters. For the camera parameters obtained by our algorithm, this maximum distance was found to be 1.6 pixels and the mean distance was 0.6 pixels.

8 Conclusion

We have presented an algorithm for camera calibration using the occluding contours of spheres. Our algorithm requires the camera to observe a sphere at three or more locations. By formulating the problem in dual space, we have shown how to recover the camera parameters optimally using semi-definite programming. The solution is exact and does not require any initialization. On the practical side, our technique is flexible, easy to use, and can be used to simultaneously calibrate multiple cameras. Therefore, our technique can find applications in calibrating multiple camera arrays for three dimensional reconstruction in meeting and lectures.

Our approach recovers the DIAC in a closed-form solution by minimizing algebraic error using semi-definite programming. However, this has no physical significance. Instead, the solution obtained by our method can be used as a starting point for a maximum likelihood estimate which minimizes the geometric error between the measured edgels and conic C over variation in the DIAC.

One of the key limitations of this approach is that the quality of the boundary fitting and ellipse detection strongly affect the accuracy of calibration results. In particular, if the sphere is imaged near the center of the image, the aspect ratio of the projected ellipse would be close to one. In this degenerate case, calibration based on spheres performs poorly. On the other hand, near the image periphery, lens distortion begins to play an important role. Hence, an analysis of the dependence of localization error of the edges on the recovered camera parameters needs to be carried out. In addition, we are looking into the effects of lens distortion on the calibration.

Acknowledgments

We thank Zhengyou Zhang for useful feedback and discussion in the early stage of the project. We are also thankful to Andrew Zisserman for some useful insights. Financial support from the "Robotics CTA" under contract # DAAD19-01-2-0012 with General Dynamics Robotics Systems is gratefully acknowledged.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

[2] P. Baker and Y. Aloimonos. Complete calibration of a multi-camera network. In *Proc. IEEE Workshop on Omnidirectional Vision*, pages 134–141, 2000.

[3] P. Beardsley, D. Murray, and A. Zisserman. Camera calibration using multiple images. In *Proc. Second European Conference Computer Vision*, pages 312–320, 1992.

[4] R. Benjemaa and F. Schmitt. A solution for the registration of multiple 3d point sets using unit quaternions. In *Proc. Fifth European Conference on Computer Vision*, pages 34–50, 1998.

[5] P. Besl and N. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[6] B. Borchers. Csdp, a c library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[7] S. Cunnington and A. Stoddart. N-view point set registration: A comparison. In *Proc. British Machine Vision Conference*, 1999.

[8] N. Daucher, M. Dhome, and J. Lapreste. Camera calibration from spheres images. In *Proc. Third European Conference on Computer Vision*, pages 449–454, 1994.

[9] L. Davis, E. Borovikov, R. Cutler, D. Harwood, and T. Horprasert. Multi-perspective analysis of human action. In *Proc. of Third International Workshop on Cooperative Distributed Vision*, 1999.

[10] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[11] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[12] R. Hartley. An algorithm for self calibration from several views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 908–912, June 1994.

[13] R. Hartley and R. Kaucic. Sensitivity of calibration to principal point position. In *Proc. Seventh European Conference on Computer Vision*, pages 433–446, 2002.

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[15] K. Kanatani. Statistical bias of conic fitting and renormalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):320–326, 1994.

[16] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[17] Y. Nesterov and A. Nemirovski. *Interior point polynomial algorithms in convex programming*. SIAM publications, Philadelphia, USA, 1994.

[18] M. Pollefeys, R. Koch, and L. van Gool. Selfcalibration and metric reconstruction inspite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32:7–25, 1999.

[19] H. Saito, S. Baba, M. Kimura, S. Vedula, and T. Kanade. Appearance-based virtual view generation of temporally-varying events from multi-camera images in the 3d room. In *Proc. of Second International Conference on 3-D Digital Imaging and Modeling*, pages 516–525, 1999.

[20] H. Teramoto and G. Xu. Camera calibration by a single image of balls: From conics to the absolute conic. In *Proc. of 5th Asian Conference on Computer Vision*, pages 499–506, 2002.

[21] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4), 1987.

[22] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[23] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[24] Z. Zhang. Camera calibration with one-dimensional objects. In *Proc. 7th European Conference on Computer Vision*, volume IV, pages 161–174, 2002.