Practical Camera Auto Calibration using Semidefinite Programming

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

Abstract

We describe a novel approach to the camera auto calibration problem. The uncalibrated camera is first moved in a static scene and feature points are matched across frames to obtain the feature tracks. Mismatches in these tracks are identified by computing the fundamental matrices between adjacent frames. The inlier feature tracks are then used to obtain a projective structure and motion of the camera using iterative perspective factorization scheme. The novelty of our approach lies in the application of semidefinite programming for recovering the camera focal lengths and the principal point. Semidefinite programming was used in our earlier work [1] to recover focal lengths under the assumption of known principal points. In this paper, we relax the constraint of known principal point and do an exhaustive search for the principal points. Moreover, we describe an end-to-end system for auto calibration and present experimental results to evaluate our approach.

1 Introduction

Over the last few years, digital video cameras have become ubiquitous because of their declining costs, thereby giving everybody easy access to video sequences. Cameras are now being deployed extensively for a wide variety of applications. Video surveillance, autonomous navigation and automotive applications are a few emerging areas in the commercialization stage. Many such applications are based on three dimensional reconstruction from the video sequences. Camera calibration is a necessary step before such reconstructions can be achieved.

In the uncalibrated setting, the best one can achieve is a reconstruction of the scene up to an unknown projective transformation. This projective structure, however, is not good enough for many applications. To obtain a euclidean reconstruction, knowledge about the calibration parameters of the camera is required. The widely used approach is to employ a calibration pattern to acquire these parameters offline. This is, however, too restrictive and time consuming. In addition, cameras may need to be recalibrated as they may loose their calibration over a period of time. Recalibrating them may not be practical and economical in many consumer applications. An automatic method that monitors the camera calibration and automatically calibrates itself would be quite appealing. This work presents a step in that direction by presenting a practical auto calibration algorithm which is well behaved and has a guaranteed convergence.

Auto calibration, recovers the camera parameters from an unknown scene using the rigidity constraints present in the scene and certain simplifying assumptions about the camera (e.g., rectangular/square pixels, known principal points, constant intrinsic parameters). Our formulation is based on the assumption that the camera has rectangular pixels (skewless camera). Our earlier work [1] assumed that the principal point of the camera is known and then recovered the focal lengths of the camera using semidefinite programming. In this work, we relax the constraint of known principal point and instead search for the principal point that minimizes a certain error measure. The error measure is based on how good the recovered focal lengths fit to the known projection matrices.

The first step in auto calibration is to recover the projection matrices and the projective structure of a camera moving in a rigid scene. Features are extracted and matched to obtain feature tracks. The mismatched features are obtained by recovering the fundamental matrices between adjacent frames of the camera. The good feature matches are used to obtain the projective structure and the camera matrices using the standard iterative perspective factorization algorithm [8]. Finally, semidefinite programming is applied to recover the aspect ratio, focal length and the principal points of the camera.

The rest of the paper is organized as follows. Section 2 describes the auto calibration problem and the various approaches for solving it. Estimation of the steps required

to estimate the projective structure is discussed in Section 4. Our formulation of the auto calibration problem in the semidefinite programming framework is discussed in Section 5. A brief introduction to semidefinite programming is provided in Section 3 and finally Section 6 presents experimental evaluation results of our approach.

Background

We assume that the reader is familiar with 3D computer vision. Standard widely used computer vision facts will be stated without derivation. Please refer to [6] for details and proofs.

Let X_i be a world point in homogenous coordinates and x_i be its projection in the camera with projection matrix P^i ; this can be represented as

$$x_j = P^i X_j \tag{1}$$

The camera matrix P^i can be decomposed as

$$P^i = \begin{bmatrix} A^i & a^i \end{bmatrix} \tag{2}$$

$$P^{i} = K^{i}R^{i} \begin{bmatrix} I - t^{i} \end{bmatrix}$$
 (3)

$$K^{i} = \begin{bmatrix} \alpha_{x} & s & x_{0} \\ & \alpha_{y} & y_{0} \\ & & 1 \end{bmatrix}$$
 (4)

$$R^{it}R^i = I (5)$$

The upper triangular matrix K^i is the matrix of intrinsic parameters and R^i and t^i are the rotation and translation of the camera respectively. Note that we are ignoring nonlinear distortion in our model. Here, α_x , α_y are the focal lengths in pixels in the x and y directions, s is the camera skew, and x_0 , y_0 are the principal points of the camera.

The absolute quadric is an imaginary quadric (i.e. with no real points) and is defined in its dual representation as a 4×4 matrix of rank 3 (Q_{∞}^*) . 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 [6] that under the camera matrix $P^{i} =$ $K^i R^i \begin{bmatrix} I & -t^i \end{bmatrix}$, the DIAC is given by $\omega^* = K^i K^{it}$, where K^{it} is the transpose of K^{i} .

The goal of auto calibration is to recover the DIAC by exploiting the rigidity constraints present in the scene. Since if ω^* is known, the intrinsic parameters K^i are easily obtained by (uniquely) finding the Cholesky factorization of ω^* , thereby updating the projective structure to a metric one. It is easy to see that the matrix ω^{*i} is symmetric and positive semidefinite (denoted by $\omega^{*i} \succeq 0$). If the skew of the camera is zero (s=0), then ω^{*i} can be written compactly as

$$\omega^{*i} = F_0 + \alpha_x^2 F_1 + \alpha_y^2 F_2 \tag{6}$$

$$F_0 = \begin{bmatrix} x_0^2 & x_0 y_0 & x_0 \\ x_0 y_0 & y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix}$$
 (7)

$$F_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

$$F_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

The earliest approach to auto calibration involved the use of Kruppa's equation [7]. These are two view constraints that require only the fundamental matrix between the two views to be known and consist of two independent quadratic equations in the elements of ω^* . Although these are the only constraints available for two views, application to three or more views does not yield satisfactory results.

The stratified approach to auto calibration [10] first recovers the plane at infinity (π_{∞}) , thereby obtaining an affine reconstruction, and then subsequently recovers the intrinsic parameter matrix. The difficulty of this approach lies in estimating π_{∞} . Methods that use scene geometry identify vanishing points of lines to determine π_{∞} . However, they do not fall strictly in the regime of auto calibration. If the internal parameters of the camera remain constant, then given three views, π_{∞} can be obtained as the intersection of three quartic equations in three variables. This is known as the modulus constraint. Because this approach requires the solution of simultaneous sets of quartic equations, very often it does not yield satisfactory results. In fact, the modulus constraint is closely related to Kruppa's constraints. Kruppa's constraints eliminates π_{∞} yielding a constraint on ω^* , whereas the modulus constraint eliminates ω^* yielding an equation in π_{∞} . Therefore, these two methods perform similarly.

The third and perhaps the most widely used approach for a large number of views solves for both ω^* and π_{∞} simultaneously. This approach was introduced by Triggs [12] in terms of an equivalent formulation using the absolute quadric (Q_{∞}) . Given n cameras, the equations relating the unknown parameters ω^{i*} and π_{∞} with the known entries of the projection matrices P^i , i = 1, ..., n are given as

$$\kappa^{i}\omega^{*i} = \left(A^{i} - a^{i}\pi_{\infty}^{t}\right)\omega^{*1}\left(A^{i} - a^{i}\pi_{\infty}^{t}\right)^{T} \quad (10)$$

$$= P^{i}Q_{\infty}^{*}P^{iT} \text{ for i=2,...,n}$$
 (11)

$$= P^{i}Q_{\infty}^{*}P^{iT} \text{ for i=2,...,n}$$

$$Q_{\infty}^{*} = \begin{bmatrix} \omega^{*1} & -\omega^{*1}\pi_{\infty} \\ -\pi_{\infty}\omega^{*1} & \pi_{\infty}\omega^{*1}\pi_{\infty} \end{bmatrix}$$
(11)

Here κ^i is an unknown scale factor. Since $\omega^{*1}, \omega^{*i} \succeq 0$, it is easy to see that the RHS is semidefinite positive (SDP) and $\kappa^i \geq 0$. Using eqn. 10, constraints on K^i are translated into constraints on ω^i that in turn gives an equation relating ω^{*1} and π_∞ . Given enough such constraints, it is possible to solve for these unknowns. When the intrinsic parameters of the camera remain constant, Triggs used a non-linear constrained minimization approach. The results of this approach are, however, strongly influenced by the initial solution. For a skewless camera with known principal point, these constraints become linear and therefore can be solved for using linear least squares (LLS) [11]. However, this approach has two major drawbacks.

- 1. The rank 3 constraint on Q_{∞}^* is not enforced in the solution. This can be later enforced in a posteriori step wherein the closest rank 3 matrix of Q_{∞}^* is used as the starting point for an iterative minimization step. However, the results are not guaranteed.
- 2. The most troublesome failing is that the positive definiteness condition of ω^* is completely ignored in the optimization step. As a result, we may end up with a ω^* that is *not* positive definite, and thus the calibration parameters cannot be recovered. The closest positive semidefinite matrix to the computed ω^* will, in most cases, not be the correct one. Hence, we have to reject this solution and start over. This situation becomes all the more prominent because of the presence of noise in the data.

Our approach overcomes these two drawbacks of the linear algorithm in a very natural manner using semidefinite programming. In practice, the solution from the linear algorithm is often used as the initial point for a non-linear approach. Our approach provides an alternative method for initialization with a guaranteed convergence and a better starting solution.

3 Semidefinite programming

Semidefinite programming (denoted SDP) is an extension of linear programming (LP) where the nonnegativity constraints are replaced by positive semidefiniteness 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, for example, the survey of Vandenberghe and Boyd [13]. Recently, SDP has also been used to calibrate a camera using spheres [2].

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

minimize
$$c^t x$$

subject to $F(x) \succeq 0$ (13)
 $F(x) = F_0 + \sum_{i=1}^m x_i F_i$

The problem data are the vector $c \in \mathbb{R}^m$ and m+1 symmetric matrices $F_0, \ldots, F_m \in \mathbb{R}^{n \times n}$. 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 semidefinite 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 [9]. 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.

3.1 Norm minimization using SDP

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

minimize
$$\zeta$$

subject to $M = \begin{bmatrix} \zeta I & A(x) \\ A(x)^t & \zeta I \end{bmatrix} \succeq 0$ (14)

If, in addition, we have a constraint that the matrix $C(x) = C_0 + x_1C_1 + \cdots + x_kC_k \succeq 0$, then this can be incorporated by replacing the constraint in eqn. 14 with

$$\operatorname{diag}(M, C(x)) \succeq 0 \tag{15}$$

Here, diag(A,B) is the block diagonal matrix with diagonal entries A and B. If B(x) is another matrix that depends linearly on x and we want to minimize $\|A(x)\| + \|B(x)\|$, then the corresponding semidefinite program to be solved is

minimize
$$\zeta + \eta$$

subject to $\operatorname{diag}\left(\left[\begin{array}{cc} \zeta I & A(x) \\ A(x)^t & \zeta I \end{array}\right], \quad (16)$

$$\left[\begin{array}{cc} \eta I & B(x) \\ B(x)^t & \eta I \end{array}\right]\right) \succeq 0$$

As before, it is easy to add constraints such as $C(x) \succeq 0$ in this framework. Generealizing this, it is easy to see that SDP can be used to minimize the sum of norms of arbitrary matrices that are affinely dependent on a variable subject to SDP constraints. In the next section, we illustrate how to formulate the auto calibration problem for any number of given cameras into a norm minimization problem.

4 Projective Reconstruction

We employ a standard method to obtain the projective reconstruction of a moving camera in a rigid scene. This is accomplished in three steps

- 1. feature detection and tracking: Harris corners [5] are detected in each frame in the video sequence. Features from each frame are matched with the features from the subsequent frame using normalized cross correlation (NCC) to give feature tracks. Since we have continuous video, a feature point can move only a fixed maximum distance between consecutive frames. For each feature point in the current frame, its NCC is evaluated for every feature point in the next frame that lies within a specified distance of its location in the current frame. This distance is taken to be 50 for our setup. Only those features that are matched to each other and with a NCC value above a threshold (taken to be 0.3) are retained. This makes the scheme more robust as only features that achieve the lowest NCC scores in each other are reliable.
- 2. fundamental matrix computation: The feature tracks obtained by the previous step are likely to have several mis matches. We remove these outliers by computing the fundamental matrix betweeen two adjacent views using a standard seven point algorithm in a RANSAC [4] framework. The feature tracks with epipolar distances above a certain threshold are discarded. Please note that this step is only used to filter the feature tracks and therefore it is not necessary to compute the multiview relations such as trifocal tensor.
- 3. iterative projective factorization: The inlier tracks that are succesfully tracked over a certain minimum number of frames are fed as input to the iterative projective reconstruction algorithm. We use the algorithm of [8] which estimates the projective depth and the scene structure alternatively in an iterative manner. Since we have removed all the incorrect matches in the previous step, the algorithm converges rapidly resulting in good projective reconstruction. In most cases the average pixel reprojection error is less than one pixel.

Given n images, the above three steps gives us the projective reconstruction with projection matrices P_i , $i=1,\ldots,n$ for each camera. Next we discuss our SDP-based formulation for the recovery of the camera calibration parameters.

5 Auto calibration using SDP

Assume for now that the principal point of the camera is known and fixed. Also, we do not require that the focal length and the aspect ratio of the camera is fixed. If the focal length of the camera is indeed constant, then the focal lengths computed from our algorithm will all be same. Hence we are given a skewless camera with known principal point and the projection matrix for each frame of the camera. Our goal is to recover α_x^i and α_y^i for each camera. Substituting the expression for ω^* (eqn.6) into the basic equation for auto-calibration (eqn.10) for the i^{th} image, we obtain

$$\kappa^{i} \left(F_{0} + \alpha_{x}^{i2} F_{1} + \alpha_{y}^{i2} F_{2} \right) = \left(A^{i} - a^{i} \pi_{\infty}^{t} \right) \left(F_{0} + \alpha_{x}^{2} F_{1} + \alpha_{y}^{2} F_{2} \right) \left(A^{i} - a^{i} \pi_{\infty}^{t} \right)^{T}$$

$$(17)$$

Let $\pi^t_\infty=\begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix}$ and $e_1,\ e_2,\ e_3$ be the three standard basis vectors for the group of 3×3 matrices (i.e., $e^t_1=\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ etc;). Let $\delta^i_0=\kappa^i,\ \delta^i_1=\kappa^i\alpha^{i2}_x$ and $\delta^i_2=\kappa^i\alpha^{i2}_y$. Then LHS can be written as

$$LHS^{i} = \sum_{j=0}^{2} \delta_{j}^{i} F_{j} \tag{18}$$

After multiplying the terms, the RHS can also be written as an affine combination of seven symmetric matrices G_0^i, \ldots, G_6^i .

$$RHS^{i} = A^{i}F_{0}A^{iT} + \alpha_{x}^{2}A^{i}F_{1}A^{iT} + \alpha_{y}^{2}A^{i}F_{2}A^{iT} + \alpha_{x}^{2}n_{1}\left(A^{i}e_{1}a^{iT} + a^{i}e_{1}^{T}A^{iT}\right) + \alpha_{y}^{2}n_{1}\left(A^{i}e_{2}a^{iT} + a^{i}e_{2}^{T}A^{iT}\right) + n_{3}\left(A^{i}e_{3}a^{iT} + a^{i}e_{3}^{T}A^{iT}\right) + \left(n_{3}^{2} + \alpha_{x}^{2}n_{1}^{2} + \alpha_{y}^{2}n_{2}^{2}\right)a^{i}a^{iT}$$

$$(10)$$

Let $\gamma_1 = \alpha_x^2$, $\gamma_2 = \alpha_y^2$, $\gamma_3 = \alpha_x^2 n_1$, $\gamma_4 = \alpha_y^2 n_2$, $\gamma_5 = n_3$ and $\gamma_6 = n_3^2 + \alpha_x^2 n_1^2 + \alpha_y^2 n_2^2$. Thus

$$RHS^{i} = G_{0}^{i} + \sum_{j=1}^{6} \gamma_{j} G_{j}^{i}$$
 (20)

Therefore, the autocalibration problem can be cast as the minimization of the sum of norm of n-1 matrices subject to certain SDP constraints.

$$minimize \sum_{i=2}^{n} \|LHS^{i} - RHS^{i}\|$$
 (21)

5.1 Constraints

Because of the parameterization that we are using, the rank constraint for Q_{∞}^* is automatically enforced. It is also

easy to add the constraint that ω_i^* is positive semidefinite.

$$F_0 + \gamma_1 F_1 + \gamma_2 F_2 \succeq 0 \tag{22}$$

$$\delta_0^i F_0 + \delta_1^i F_1 + \delta_2^i F_2 \succeq 0 \quad i = 2, \dots, n$$
 (23)

The expressions for γ_1 , γ_2 and γ_6 imply that all of these are non-negative. Similarly, δ_0^i , δ_1^i and δ_2^i are also nonnegative. Therefore, we have

diag
$$(\gamma_1, \gamma_2, \gamma_6, \delta_0^2, \dots, \delta_0^n, \delta_1^2, \dots, \delta_1^n, \delta_2^2, \dots, \delta_2^n) \succeq 0$$
(24)

The above SDP constraints along with the norm minimization of the sum in eqn. 21 make a constrained norm minimization problem. The variables in this case are $\gamma_1, \ldots,$ $\gamma_6, \, \delta_0^i, \, \delta_1^i, \, \delta_2^i$ for $i=2,\ldots,n$. SDP can then be used to solve this minimization problem and obtain the gammas and deltas from which the focal length of each camera can then be obtained. In addition the plane at infinity π_{∞} is also recovered.

5.2Approximation error

It is easy to see that

$$\gamma_6 = \gamma_5^2 + \frac{\gamma_3^2}{\gamma_1} + \frac{\gamma_4^2}{\gamma_2} \tag{25}$$

Therefore all the gammas are not independent. There is no way to incorporate this non convex constraint into the SDP formulation. Therefore, the SDP formulation described above is not quite exact. It is only a convex relaxation of the original non convex optimization problem. However barring this constraint on γ_6 , our SDP formulation is an exact representation of the original auto calibration problem and yields satisfactory results.

For most cases, the focal length of the camera will be the same for all the frames in the sequence. In that case we must have $\alpha_x^{i2}=\alpha_x^2$ and $\alpha_y^{i2}=\alpha_y^2$ for $i=2,\ldots,n$. Rewriting these in terms of gammas and deltas, we have

$$\delta_1^i = \delta_0^i \gamma_1 \qquad (26)$$

$$\delta_2^i = \delta_0^i \gamma_2 \qquad (27)$$

$$\delta_2^i = \delta_0^i \gamma_2 \tag{27}$$

These are non convex constraints and these constraints can not be directly incorporated into the SDP formulation. Thus, for the constant focal lengths caseour formulation also solves a relaxed version of the original problem. However, for most cases the semidefinite constraints in the above formulation are very strong and in practice, we have found that the recovered focal lengths are almost always nearly identical.

5.3 Search for the principal points

The SDP formulation described above assumes that the principal point is known and fixed. The principal point of the camera is known to be notoriously hard to compute. Often, therefore, the principal point position is guessed to be at the center of the image. We assume that the principal point is fixed and perform an exhaustive search for the principal point near the center of the image. For each principal point location (x_0, y_0) , we formulate the SDP based auto calibration problem and solve for the focal lengths and the plane at infinity. The residual error of the sum of the matrix norms in equation 21 gives an estimate of how good a particular point location is. This error is simply computed from the SDP formulation as the residual error upon convergence. %endequation The location (x_0, y_0) resulting in the minimum total residual error is taken as the principal point of the camera and the corresponding focal lengths estimated from the SDP formulation for that principal point.

For computational efficiency, we perform a stratified exhaustive search on the principal point position near the center of the image. We start with larger increments of x_0 and y_0 in the image and subsequently refine the search for the principal point to a pixel. Our experiments, using our approach indicate that the inaccuaracies in the principal point do not have a significant effect on the other parameters of the camera.

6 Results

We have implemented the entire algorithm efficiently. We have used CSDP [3], a publicly available package for solving semidefinite programming problems.

We have applied our algorithm for both simulated data and real images. For the simulations, a random cloud of 500 3D points was generated. Images of these points were obtained in a sequence of 10 frames by moving the camera and changing its focal lengths while keeping the principal point fixed. Random gaussian noise was then added to the location of the projected points. Table 1 shows the recovered focal lengths for frames 1, 6 and 10 and principal points using our algorithm(SDP) for noise with $\sigma = 0.5, 1.0, 2.0$ pixels ($\sigma = 0$ corresponds to ground truth). The search area for the principal point was 50 pixels in the x and y direction around the center of the image.

In our second experiment, a camera with fixed parameters was moved in a rigid indoor scene. Feature points were obtained and tracked in ten frames of this sequence of images, each of size 640×480 . A projective reconstruction was obtained using these correspondences, and our algorithm was applied to obtain the camera parameters. Since the camera parameters were fixed, the recovered focal lengths should be constant. However, due to noise these focal lengths will vary. The mean focal lengths obtained for these 10 frames was $\alpha_x = 888$ and $\alpha_y = 893$. The standard deviation for the focal lengths was $sd_x = 3.6$ and $sd_y = 3.4$, indicating that the recovered focal lengths did Table 1. Results for simulated data

Table II Hobalto Ioi Cillialatea aata				
Cam	$\sigma = 0$	$\sigma = 0.5$	$\sigma = 1.0$	$\sigma = 2.0$
x_0	400	403	376	368
y_0	300	320	332	332
α_x^1	666	673	657	589
α_y^1	673	691	634	602
α_x^6	629	641	637	583
$\begin{array}{c} \alpha_x^6 \\ \alpha_y^6 \end{array}$	633	654	605	575
α_x^{10}	603	607	614	558
α_y^{10}	649	667	604	584

not vary much. The principal point resulting in minimum residual error was $x_0=296$ and $y_0=227$. The camera was also calibrated offline using a plane calibration grid yielding focal lengths $\alpha_x=928$, $\alpha_y=926$ and principal points $x_0=327$ and $y_0=236$. It is easy to see that our results agree closely with the results from the plane based calibration method which is considered to be the "goldstandard" method for calibration.

7 Conclusion

We have presented a novel approach for the auto calibration of a camera using semidefinite programming. Our approach recovers the aspect ratio, focal length and the principal point of a camera moving rigidly in a static scene. Our approach naturally incorporates the constraint that ω^* is positive semidefinite and therefore is much more stable. Although our approach exhaustively searches for the principal point of the camera resulting in the minimum residual error, it has been our experience that this point can often be assumed to be at or near the center of the image without much impact on the computed focal lengths. Perhaps, one of the biggest enhancements to this work would be to also estimate the camera distortion parameters simultaneously. For most narrow field lenses, camera distortion is not significant. We are currently working on solving for the camera distortion parameters to make this approach more general and practical.

References

- [1] M. Agrawal. On automatic determination of varying focal lengths using semidefinite programming. In *Proceedings IEEE International Conference on Image Processing (ICIP)*, 2004.
- [2] M. Agrawal and L. Davis. Camera calibration using spheres: A semidefinite programming approach. In *ICCV*, pages 782–789, 2003.
- [3] B. Borchers. Csdp, a c library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

- [4] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. ACM.*, 24:381– 395, 1981.
- [5] C. Harris and M. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–151, 1988.
- [6] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.
- [7] Q. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *IJCV*, 22(3):261–289, 1997.
- [8] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provablyconvergent iterative methods for projective structure from motion. In *Proc. CVPR*, 2001.
- [9] Y. Nesterov and A. Nemirovski. *Interior point polynomial algorithms in convex programming*. SIAM publications, Philadelphia, USA, 1994.
- [10] M. Pollefeys and L. V. Gool. A stratified approach to metric self-calibration. In *Proc. CVPR*, pages 407–412, 1997.
- [11] M. Pollefeys, R. Koch, and L. van Gool. Self-calibration and metric reconstruction inspite of varying and unknown internal camera parameters. In *Proc. ICCV*, pages 90–95, 1998
- [12] B. Triggs. Autocalibration and the absolute quadric. In *Proc. CVPR*, pages 609–614, 1997.
- [13] L. Vandenberghe and S. Boyd. Semidefinite programming. SIAM Review, 38(1):49–95, 1996.