

Self-Calibrating Cameras Using Semidefinite Programming

Chunhua Shen^{1,2}, Hongdong Li^{2,1}, Michael J. Brooks³

¹NICTA Canberra Research Lab ²Australian National University ³University of Adelaide

Abstract

Novel methods are proposed for self-calibrating a pure-rotating camera using semidefinite programming (SDP). Key to the approach is the use of the positive-definiteness requirement for the dual image of the absolute conic (DIAC). The problem is couched within a convex optimization framework and convergence to the global optimum is guaranteed. Experiments on various data sets indicate that the proposed algorithms more reliably deliver accurate and meaningful results. This work points the way to an alternative and more general approach to self-calibration using the advantageous properties of SDP. Algorithms are also discussed for cameras undergoing general motion.

1. Introduction

Self-calibration is the process by which camera parameters are automatically determined from image point data that has been matched across multiple images [17]. It has proven a remarkable development in computer vision in that it has often obviated the need in vision applications for a tedious prerequisite step involving a known calibration pattern. Elemental in the formulation of the self-calibration problem are Kruppa's equations [12], and some early self-calibration techniques were based directly upon solving these equations [17]. However, their use has not subsequently found favor as these techniques are susceptible to noise and often fail to produce physically meaningful results [15]. A discussion of the difficulties associated with methods based upon Kruppa's equation can be found in [16]. Favored techniques for self-calibration have been the linear method [8], based upon simple algebraic least squares, and nonlinear iterative methods, such as bundle adjustment [18]. It emerges that a key constraint cannot be incorporated by these methods. Furthermore, the bundle-adjustment approach is typically highly sensitive to initialization accuracy. In this paper, we seek to overcome the disadvantages associated with the direct use of Kruppa's equations. In the context of self-calibrating a rotating camera, we develop novel techniques cast within a

semidefinite programming (SDP) framework. In this way, the positive-definiteness constraint of the dual image of absolute conic (DIAC) is fully incorporated and global convergence is guaranteed. This paper further develops work presented in [14].

1.1. Related work

For a general overview of camera self-calibration, the interested reader is referred to [12]. The work most closely related to this paper is due to Agrawal *et al.* [3, 2]. Here use is made of the positive-definiteness constraint of the DIAC. In contrast with our formulation, however, Agrawal [2] estimates only the focal length and assumes other parameters are known. The optical center is assumed coincident with the optical center of the camera, and this is known to be problematical. Indeed, Hartley and Kaucic [10] have shown that errors in the assumed optical center can have a deleterious influence on the estimated focal length of the cameras. In [3] three images of a sphere are used to calibrate a camera. Both of these papers use the spectral matrix norm as the cost function. In contrast with these works, for the rotating camera self-calibration problem, we utilize the Kruppa's equations, estimate all camera parameters, adopt three possible norms and devise a semidefinite programming framework that guarantees convergence to the global optimum.

A global optimization method, termed interval analysis, has been applied to camera self-calibration by Benedetti *et al.* [4]. Like many other global methods, the computational complexity of interval analysis is very high. Moreover the algorithm is not easy to implement. Since [11] introduced ℓ_∞ minimization into multi-view geometry, there has been a growing interest in seeking a globally optimal solution using mathematical programming tools. For example, [13] hierarchically convexifies nonconvex optimization problems with polynomials. At each step an SDP is solved. In [1] several geometry problems are formulated as fractional programs and a branch and bound procedure is implemented to find the global optimum.

Notation. The (m, n) entry of a matrix X is represented by X_{mn} . The trace of X is denoted by $\text{Tr}(X)$. The operator $\text{vec}(X)$ stacks all the columns of the matrix X into a column vector. The inner product of two

matrices is defined as $\langle X, Y \rangle = \text{Tr}(X^T Y)$. The spectral norm (*a.k.a.* matrix norm) of a matrix X , denoted by $\|X\|_2$, is the square root of the largest eigenvalue of $X^T X$, *i.e.*, $\|X\|_2 = \sqrt{\lambda_{\max}(X^T X)}$. The Frobenius norm of X is given by: $\|X\|_F = \sqrt{\sum_{mn} X_{mn}^2}$. The ℓ_1 norm of X is defined as $\|X\|_1 = \sum_{mn} |X_{mn}|$. A positive semidefinite (p.s.d.) matrix X is represented by $X \succcurlyeq 0$. For a p.s.d. matrix X , we have $\mathbf{v}^T X \mathbf{v} \geq 0, \forall \mathbf{v}$.

2. Rotating cameras

We consider the self-calibration of cameras with pure rotation. An algorithm for calibrating a rotating camera was given in [8] and was generalized in [7] to allow for zooming and possible variation of the optical center of the camera. We briefly summarize below the linear calibration method given in [7]. Assume that we have a set of images taken with cameras located at the same point in space, which is defined as the coordinate origin. The internal matrix, K^i , of each camera is a 3×3 matrix given by $K^i = [f \ \kappa f \ u_0; 0 \ \alpha f \ v_0; 0 \ 0 \ 1]$, where f is the focal length, α is the aspect ratio, u_0 and v_0 are the coordinates of the optical center and κ is the skew parameter.

For a camera undergoing pure rotation, any two images of a static scene are related by a planar homography, which is a pure geometric relationship independent of scene contents. We denote the homography between the i th and j th images as H^{ij} . The homographies can be calculated by direct measurement of matching points in the set of images [12]. Denoting by X^i the DIAC in the i th image, the following simplified form of the Kruppa equations defines the rules for transforming conics under homographies: $X^i = H^{ij} X^j (H^{ij})^T$. The DIAC is related to the camera's internal matrix by the formula $X^i = K^i (K^i)^T$. These equations serve as the basis for camera self-calibration. The calibration matrix K^i can be easily computed from X^i by Cholesky decomposition. Specifically any symmetric positive semidefinite matrix can be uniquely factored as product of $K K^T$ such that K is an upper triangular matrix with positive diagonal entries. Clearly X^i must be positive semidefinite.

2.1. Semidefinite programming

We briefly recall some important results on semidefinite programming (SDP). These results can be found in [5]. The SDP approach minimizes a linear objective under a linear matrix inequality (LMI) constraint. An LMI is a constraint on a vector $\mathbf{x} = \{x_1, x_2, \dots, x_m\}$ of the form $\mathcal{F}(\mathbf{x}) = \mathcal{F}_0 + \sum_i^m x_i \mathcal{F}_i \succcurlyeq 0$, where $\mathcal{F}_i = \mathcal{F}_i^T, i = 0, 1, \dots, m$ are symmetric matrices. SDP optimization is a special case of the more general convex programming, which has the

standard form

$$\text{minimize } \mathbf{c}^T \mathbf{x}, \quad \text{s.t. } \mathcal{F}(\mathbf{x}) \succcurlyeq 0.$$

Here \mathbf{c} is a vector that models the problem at hand. Note that multiple LMIs and linear inequalities can be included and transformed into the standard form of SDP [5]. An SDP is convex, hence global optimality is guaranteed, and can be solved in polynomial time with interior-point methods. It is observed that the number of iterations of interior-point methods is almost constant, independently of problem size. The problems we are interested in here are small and can be solved very efficiently. Off-the-shelf SDP software packages are available, *e.g.*, SeDuMi and CSDP. If the matrices \mathcal{F}_i are all diagonal, then the LMI is a set of linear inequalities, and the SDP reduces to a linear program. Indeed, several optimization techniques, such as second-order cone programming (SOCP) and quadratically-constrained quadratic programming (QCQP), are special cases of SDP. An extremely useful tool for converting a problem into SDP forms is the Schur complement method described below.

Theorem 2.1. *Consider a symmetric matrix X partitioned as $X = [A \ B; B^T \ C]$, where A is symmetric. If $\det(A) \neq 0$, the matrix $S = C - B^T A^{-1} B$ is the Schur complement of A in X . Then we have: (1) $X \succ 0$ if and only if $A \succ 0$ and $S \succ 0$; (2) If $A \succ 0$, then $X \succcurlyeq 0$ if and only if $S \succcurlyeq 0$.*

See [5] for the proof. Next we show how camera calibration under various cost functions can be cast as semidefinite programming problems.

2.2. SDP formulation for constant internal parameters

We consider the case that the camera only undergoes a two-axis rotation, and no zooming is allowed. In this case the camera's internal parameters remain constant throughout the sequence. Under this assumption, camera self-calibration is achieved by solving the following equations¹

$$X \simeq H^{ij} X (H^{ij})^T, \quad (1)$$

where \simeq denotes equality up to a nonzero scalar factor. Given this scale ambiguity, the entry (3, 3) of K can be assumed to be 1. It is easy to see that the constraint $K_{33} = 1$ is equivalent to $X_{33} = 1$. For simplicity, we describe our algorithm for three image frames. However, the presented algorithm can be trivially extended to arbitrarily many views in a straightforward way. Inevitably, with noise in the measurement, and hence the homographies, it is not possible that Equation (1) will be exactly satisfied. Instead we wish

¹The superscripts of X are dropped because we are considering the case that the internal calibration matrix is constant.

to minimize a cost function under some certain criterion. Mathematically, we want to solve:

$$\text{minimize } F(X, H^{ij}X(H^{ij})^\top) \text{ s.t. } X \succcurlyeq 0, X_{33} = 1. \quad (2)$$

Here, $F(\cdot, \cdot)$ measures the proximity of two matrices. How should this proximity be measured? There are many metrics defined in matrix analysis. Typical choices are the ℓ_1 , Frobenius and spectral norms.

2.2.1 The ℓ_1 norm cost

The ℓ_1 norm is one of the most widely used metrics to measure the distance between vectors or matrices. In the case of the ℓ_1 norm, we want to minimize

$$\|X - H^{12}X(H^{12})^\top\|_1 + \|X - H^{13}X(H^{13})^\top\|_1.$$

For ease of presentation we define

$$Z^{12} = X - H^{12}X(H^{12})^\top, \quad (3)$$

$$Z^{13} = X - H^{13}X(H^{13})^\top. \quad (4)$$

Both Z^{12} and Z^{13} are linear in X . Introducing the auxiliary variables δ_{mn} and δ'_{mn} ($m, n = 1, 2, 3$; and $m \leq n$). The optimization problem is then equivalent to

$$\begin{aligned} & \text{minimize}_{X, \delta_{mn}, \delta'_{mn}} \sum_{m, n=1, 2, 3; m \leq n} (\delta_{mn} + \delta'_{mn}) \\ & \text{s.t. } \delta_{mn} \geq 0, \delta'_{mn} \geq 0, \\ & \quad -\delta_{mn} \leq Z_{mn}^{12} \leq \delta_{mn}, \\ & \quad -\delta'_{mn} \leq Z_{mn}^{13} \leq \delta'_{mn}, (m, n = 1, 2, 3; m \leq n) \\ & \quad X \succcurlyeq 0, X_{33} = 1. \end{aligned} \quad (5)$$

Here, we have converted the ℓ_1 norm optimization into an SDP since all the constraints are linear except for the LMI. Note that the summation above can be restricted to the range $m \leq n$, since the involved matrices are symmetric.

2.2.2 The Frobenius norm cost

The Frobenius norm is of interest because almost all the previous rotating camera self-calibration algorithms minimize this particular least-squares cost function [8, 7] which can be expressed as

$$\|X - H^{12}X(H^{12})^\top\|_F^2 + \|X - H^{13}X(H^{13})^\top\|_F^2.$$

This is actually a (generalized) semidefinite least-squares (SDLS) problem [6]. Methods exist for solving large-scaled SDLS problems. Nevertheless for camera calibration, the problem size is relatively very small. Therefore in this work

we formulate this generalized SDLS problem as an SDP. We rewrite the original problem as

$$\begin{aligned} & \text{minimize}_{X, \delta_1, \delta_2} (\delta_1 + \delta_2) \\ & \text{s.t. } \|X - H^{12}X(H^{12})^\top\|_F^2 \leq \delta_1 \\ & \quad \|X - H^{13}X(H^{13})^\top\|_F^2 \leq \delta_2 \\ & \quad X \succcurlyeq 0, X_{33} = 1. \end{aligned} \quad (6)$$

Recalling Z^{12} in (3), the first constraint in (6) is equivalent to $[\text{vec}(Z^{12})]^\top [\text{vec}(Z^{12})] \leq \delta_1$. Again, with the Schur complement lemma this constraint can be expressed as an LMI:

$$[\delta_1 \quad [\text{vec}(Z^{12})]^\top; \quad \text{vec}(Z^{12}) \quad \mathbf{I}] \succcurlyeq 0.$$

Here \mathbf{I} is an identity matrix. We also have

$$[\delta_2 \quad [\text{vec}(Z^{13})]^\top; \quad \text{vec}(Z^{13}) \quad \mathbf{I}] \succcurlyeq 0.$$

for the second constraint. We have thus converted the Frobenius cost minimization into an SDP. These two constraints are second-order cones. Unfortunately it is impossible to convert (6) into an SOCP due to the p.s.d. constraint $X \succcurlyeq 0$.

2.2.3 The spectral norm cost

The spectral norm of a symmetric matrix is its largest eigenvalue value. Obviously when the spectral norm is close to zero, each entry of the matrix approaches zero. In this case, the cost function becomes

$$\|X - H^{12}X(H^{12})^\top\|_2 + \|X - H^{13}X(H^{13})^\top\|_2.$$

By introducing two non-negative auxiliary variables δ_1 and δ_2 , we rewrite the optimization problem as:

$$\begin{aligned} & \text{minimize}_{X, \delta_1, \delta_2} (\delta_1 + \delta_2) \\ & \text{s.t. } \|X - H^{12}X(H^{12})^\top\|_2 \leq \delta_1 \\ & \quad \|X - H^{13}X(H^{13})^\top\|_2 \leq \delta_2 \\ & \quad X \succcurlyeq 0, X_{33} = 1. \end{aligned} \quad (7)$$

With the Schur complement lemma, the first two constraints can be converted into LMIs as follows

$$[\delta_1 \mathbf{I} \quad (XH^{12}X(H^{12})^\top)^\top; \quad XH^{12}X(H^{12})^\top \quad \mathbf{I}] \succcurlyeq 0,$$

and

$$[\delta_2 \mathbf{I} \quad (XH^{13}X(H^{13})^\top)^\top; \quad XH^{13}X(H^{13})^\top \quad \mathbf{I}] \succcurlyeq 0.$$

Now the first three constraints are all LMIs and the last one is a linear equality. Furthermore, the cost function is linear, so the problem is an SDP.

3. Experiments

We now test the SDP algorithms on synthetic image data and compare the results with those of the linear method.

Simulated data are generated as follows. A set of 200 uniformly random 3D points is distributed within in a cube and sampled. The 3D points are projected onto images via a perspective camera. Multiple images are generated via pure rotation of the camera. The simulated image size is 256×256 , and image coordinates are corrupted by Gaussian noise with various levels of standard deviation.

In the first experiment, we compare the accuracy of focal length estimation. The experiments are repeated over 150 runs and the average error is presented. For the proposed SDP algorithms, the three different norms give similar results. Figure 1 (left) plots the error of the focal length with SDP (ℓ_1 norm) and the standard linear algorithm. The linear algorithm gives the optimal solution in terms of accuracy but drop the requirement of positive-definiteness of K . The errors for the SDP algorithm are very close to that obtained by the linear algorithm. In theory, our SDP formulation should have the same accuracy as the linear methods. In practice, because of the SDP solver's implementation, there are numerical artifacts. Also since our SDP adopts the interior-point Newton algorithm which solves the problem iteratively, to set the stopping threshold smaller improves the accuracy. But the price is more iterations needed for convergence. Also evaluated was the success rate-that is, the percentage that the estimated X is indeed positive definite and a valid K can be attained. Figure 1 (right) shows that SDP always produces a numerically meaningful estimate, in contrast with the linear method.

Figures 2 and 3 summarize the results. Table 1 shows the results of SDP (ℓ_1) at different levels of noise and Table 2 reports the results obtained by different methods under for a noise level of 0.4 pixels.

Again it can be seen from Figure 3 that SDP algorithms give close solutions to the linear method. At the same time they are more reliable.

4. Globally Solving the Kruppa's Equations

We have discussed how to self-calibrate rotating cameras using SDPs, which is based on the reduced form of the Kruppa's equations. We discuss how to solve the Kruppa's equation globally in this section such that we can deal with general camera motions. Most practical self-calibration approaches based on Kruppa's equations depend on the epipolar geometry of pairs of views and identify the IAC. Kruppa's equations can be viewed as an epipolar matching constraint for the projections of quadratics or conics. There are various ways to obtain the Kruppa's equations, *e.g.*, by utilizing algebraic relationships between projective geometric

noise	f	αf	κf	u_0	v_0
–	800	800	160	10	20
0.1	785.2	785.2	157.1	10.4	19.9
0.18	778.0	777.7	155.7	11.7	20.6
0.26	748.3	748.2	149.2	12.5	22.3
0.4	760.9	760.5	152.4	10.5	20.6
0.6	724.1	724.9	146.0	10.6	25.0
0.8	692.2	692.8	137.5	14.3	26.7

Figure 2: Calibration results of SDP (ℓ_1 norm). The first row shows ground truth, and subsequent rows show the effects of different level of noise (measured in pixels).

method	f	αf	κf	u_0	v_0
ground truth	700	770	140	0	0
SDP (ℓ_1)	643.2	708.6	130.6	-1.3	2.5
SDP (Frobenius)	636.8	701.0	128.2	-0.8	3.3
SDP (spectral)	624.5	688.4	126.1	-0.5	5.5
linear	658.7	764.3	121.5	-1.2	0.6

Figure 3: Calibration results of different methods. The noise level is 0.4 pixels. The first row shows ground truth. For the linear method, 4 trials out of 150 fail to give a valid estimate.

quantities or by singular value decomposing (SVD) of the fundamental matrix.

If we let $Y = KK^T = [p \ q \ r; q \ s \ t; r \ t \ 1] \succcurlyeq 0$, with K is camera's internal matrix, the explicit Kruppa's equations are:

$$\frac{\mathbf{a}_1^T Y \mathbf{a}_1}{\mathbf{a}_1^T Y \mathbf{a}'_1} = \frac{\mathbf{a}_1^T Y \mathbf{a}_2}{\mathbf{a}_1^T Y \mathbf{a}'_2} = \frac{\mathbf{a}_2^T Y \mathbf{a}_2}{\mathbf{a}_2^T Y \mathbf{a}'_2}, \quad (8)$$

where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}'_1, \mathbf{a}'_2 \in \mathbb{R}^{3 \times 1}$ are column vectors. They can be found by an SVD of the fundamental matrix [9]. Obviously there are two forms for Kruppa's equations: the polynomial form and the fractional form. We use the polynomial form of the Kruppas equations. Each motion then provides three polynomials and two of them are independent. Therefore the cost function we wish to minimize is:

$$F(Y) = \sum_i (|u_{11}^i u_{11}^i - u_{11}^i u_{12}^i| + |u_{12}^i u_{22}^i - u_{12}^i u_{22}^i| + |u_{11}^i u_{22}^i - u_{11}^i u_{22}^i|). \quad (9)$$

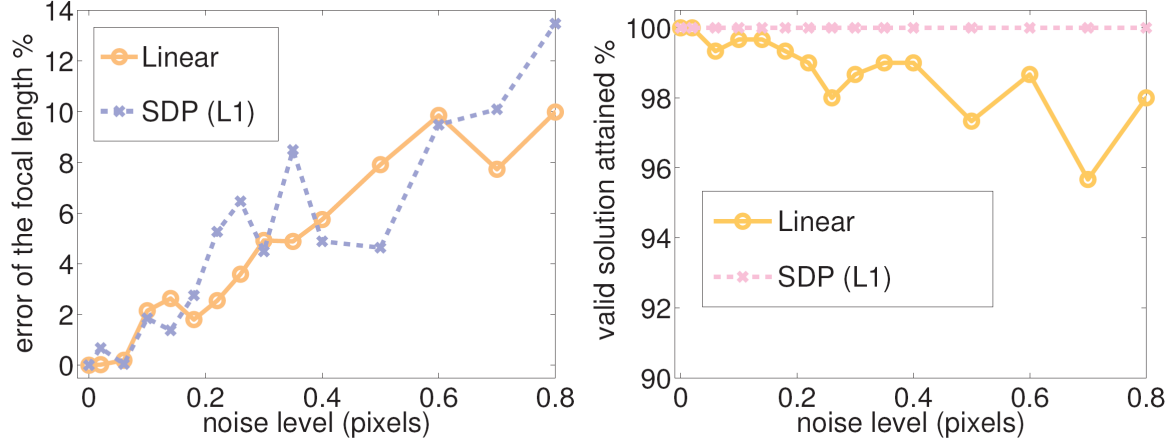


Figure 1: Calibrating a rotating camera from three images using SDP (ℓ_1 norm) and the standard linear method in the presence of 16 different degrees of noise. The size of the image is 256×256 pixels. (left) Relative errors of the focal length estimated. (right) Rate of valid solutions attained. SDP always outputs a valid solution.

Here the superscript i indexes the motion; $|u|$ is the absolute value of u . We have also defined the following shorthand symbols: $u_{11} = \mathbf{a}_1^\top Y \mathbf{a}_1$, $u'_{11} = \mathbf{a}'_1^\top Y \mathbf{a}'_1$, $u_{12} = \mathbf{a}_1^\top Y \mathbf{a}_2$, $u'_{12} = \mathbf{a}'_1^\top Y \mathbf{a}'_2$, $u_{22} = \mathbf{a}_2^\top Y \mathbf{a}_2$, $u'_{22} = \mathbf{a}'_2^\top Y \mathbf{a}'_2$. They are all linear in Y . Typically the above absolute operation (ℓ_1 criterion) can be replaced with the least-squares criterion, which yields a smooth cost function and iterative gradient methods (e.g., the Levenberg-Marquardt algorithm) can be applied.

Using similar techniques as solving the ℓ_1 -norm cost function of the rotating camera, the optimization is formally

$$\begin{aligned}
 & \underset{Y, \delta_0^i, \delta_1^i, \delta_2^i}{\text{minimize}} && \sum_i (\delta_0^i + \delta_1^i + \delta_2^i) \\
 & \text{subject to} && -\delta_0^i \leq u_{11}^i u_{11}^i - u_{11}^i u_{12}^i \leq \delta_0^i, \\
 & && -\delta_1^i \leq u_{12}^i u_{22}^i - u_{12}^i u_{11}^i \leq \delta_1^i, \\
 & && -\delta_2^i \leq u_{11}^i u_{22}^i - u_{11}^i u_{12}^i \leq \delta_2^i \quad \forall i, \\
 & && Y \succcurlyeq 0.
 \end{aligned} \tag{10}$$

Same as previous, here $\delta_0^i, \delta_1^i, \delta_2^i$ are auxiliary variables.

Equation (10) is not convex and hence difficult to solve. We now show how to convert this problem into an SDP.

First we define a column vector $\mathbf{z} = [p \ q \ r \ s \ t \ 1]^\top$, which is the vectorized upper triangle of Y , and we have $Z = \mathbf{z}\mathbf{z}^\top \succcurlyeq 0$. Clearly we also have $\text{rank}(Z) = 1$. In terms of Z , the constraint $Y \succcurlyeq 0$ in (10) can be expressed as

$$\begin{bmatrix} Z_{16} & Z_{26} & Z_{36} \\ Z_{26} & Z_{46} & Z_{56} \\ Z_{36} & Z_{56} & 1 \end{bmatrix} \succcurlyeq 0.$$

It is critically important to observe that $u_{11}^i u_{11}^i - u_{11}^i u_{12}^i$ (and the other two) can be written as a linear expression

of Z . For ease of presentation, we introduce matrices $A^i, B^i, C^i \in \mathbb{R}^{6 \times 6}$. The non-convex constraints in (10) are equivalent to

$$-\delta_0^i \leq \langle A^i, Z \rangle \leq \delta_0^i, \tag{11}$$

$$-\delta_1^i \leq \langle B^i, Z \rangle \leq \delta_1^i, \tag{12}$$

$$-\delta_2^i \leq \langle C^i, Z \rangle \leq \delta_2^i \quad (\forall i). \tag{13}$$

Since it is difficult to directly solve for Y , we instead solve for Z and it is trivial to obtain Y from Z . By putting the above analysis together, the problem (10) is equivalent to:

$$\begin{aligned}
 & \underset{Z, \delta_0^i, \delta_1^i, \delta_2^i}{\text{minimize}} && \sum_i (\delta_0^i + \delta_1^i + \delta_2^i) \\
 & \text{subject to} && (11), (12), (13) \\
 & && Z \succcurlyeq 0, \text{rank}(Z) = 1.
 \end{aligned} \tag{14}$$

The equivalence can be easily established. The above optimization problem is still non-convex due to the rank constraint. The seemingly simple rank constraint leads to an NP-hard optimization problem. Rank-constrained LMIs frequently arise in mathematical programming. As yet there is no effective solution method. If we drop the rank constraint $\text{rank}(Z) = 1$, we can form a relaxation the original problem, which is an SDP in the variables Z and $\delta_0^i, \delta_1^i, \delta_2^i$. The optimal value of the relaxed program will be a lower bound on the optimal value of the original problem.

Because of the relaxation, the matrix rank constraint is removed and therefore the SDP solution might be lifted to a higher rank (higher dimensional space). In this higher dimensional space, the cost function is lower than it would be

when the solution is constrained to rank one. An important SDP research topic is how to round the higher-dimension (higher rank) SDP solution into a lower-dimension (desired low rank) solution. In our case, a naive treatment is to project the solution Z to the rank-one semidefinite cone. A nearest suboptimal Z^* is then obtained. This can be easily performed using the eigenvalue decomposition. Similar ideas are used to compute a low-rank embedding from the top eigenvectors in maximum variance folding [19]. This suboptimal Z^* can then be used as an initial starting position for Levenberg-Marquardt bundle adjustment refinement. With a reasonable initial guess, a local search can be efficient for finding a low-rank solution in many cases.

5. Conclusion

We have shown how to transform the rotating camera self-calibration problem into a semidefinite programming problem. In doing so, the positive-definiteness constraint is seamlessly integrated. Because the reformulated problem involves convex optimization, meaningful solutions are guaranteed, in contrast with the operation of the linear algorithm. This work points the way to a more general approach to self calibration for which accurate and meaningful solutions are assured.

Acknowledgments

NICTA is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

- [1] S. Agarwal, M. Chandraker, F. Kahl, S. Belongie, and D. J. Kriegman. Practical global optimization for multiview geometry. In *Proc. Eur. Conf. Comp. Vis.*, volume 1, pages 592–605, Graz, Austria, 2006.
- [2] M. Agrawal. On automatic determination of varying focal lengths using semidefinite programming. In *Proc. IEEE Int. Conf. Image Process.*, pages 3379–3382, Singapore, 2004.
- [3] M. Agrawal and L. S. Davis. Camera calibration using spheres: A semi-definite programming approach. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 782–789, Nice, France, 2003.
- [4] A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using interval analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(12):1633–1638, 2004.
- [5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [6] S. Boyd and L. Xiao. Least-squares covariance matrix adjustment. *SIAM J. Matrix Anal. Appl.*, 27(2):532–546, 2005.
- [7] L. de Agapito, E. Hayman, and I. Reid. Self-calibration of rotating and zooming cameras. *Int. J. Comp. Vis.*, 45(2):107–127, 2001.
- [8] R. I. Hartley. Self-calibration from multiple views with a rotating camera. In *Proc. Eur. Conf. Comp. Vis.*, volume 1, pages 471–478, Stockholm, Sweden, 1994. Springer-Verlag.
- [9] R. I. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):133–135, 1997.
- [10] R. I. Hartley and R. Kaucic. Sensitivity of calibration to principal point position. In *Proc. Eur. Conf. Comp. Vis.*, volume 2, pages 433–446, Copenhagen, Denmark, 2002.
- [11] R. I. Hartley and F. Schaffalitzky. ℓ_∞ minimization in geometric reconstruction problems. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, volume 1, pages 504–509, Washington, DC, 2004.
- [12] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.
- [13] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. In *Proc. IEEE Int. Conf. Comp. Vis.*, volume 2, pages 978–985, Beijing, China, 2005.
- [14] H. Li and C. Shen. An LMI approach for reliable PTZ camera self-calibration. In *Proc. IEEE Int. Conf. Advanced Video & Signal Based Surveillance*, pages 79–84, Sydney, Australia, 2006.
- [15] Q.-T. Luong and O. D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *Int. J. Comp. Vis.*, 22(3):261–289, 1997.
- [16] Y. Ma, R. Vidal, J. Kosecka, and S. Sastry. Kruppa equation revisited: Its renormalization and degeneracy. In *Proc. Eur. Conf. Comp. Vis.*, volume 2, pages 561–577. Springer-Verlag, 2000.
- [17] S. J. Maybank and O. D. Faugeras. A theory of self calibration of a moving camera. *Int. J. Comp. Vis.*, 8(2):123–151, 1992.
- [18] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment — a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.
- [19] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comp. Vis.*, 70(1):77–90, 2006.