

An LMI Approach for Reliable PTZ Camera Self-Calibration

Hongdong Li¹, Chunhua Shen²
RSISE, The Australian National University¹
ViSTA, National ICT Australia, Canberra Labs².

Abstract

PTZ (Pan-Tilt-Zoom) cameras are widely used for large-area video surveillance. For many visual tracking and video analysis tasks, an accurate camera calibration is very important. Traditional off-line camera calibration algorithms are often not satisfactory because some of the PTZ camera intrinsic parameters (e.g., focal length) may change during working. Despite of their theoretical elegance, existing on-line camera self-calibration algorithms are not satisfactory either, because of the lack of numerical stability. This paper proposes a new method based on LMI (linear matrix inequality) optimization. This method automatically incorporates the required positive-definiteness constraint into the computation, thus delivers more reliable and more stable results. Experiments on both synthetic data and real images have validated the advantages of our method.

1. Introduction

Modern video surveillance systems are expected to have more intelligence than the conventional passive video monitoring systems have. For example, such system is often demanded to have the ability of analyzing video contents, segmenting objects, tracking moving targets, measuring 3D geometric and kinetic (i.e, speed) information, and understanding visual events, etc. For all these *intelligent* capabilities, accurate *camera calibration* is very essential.

PTZ (Pan-Tilt-Zoom) camera is very popular in video surveillance of various environments (see figure-1, [12]) because it can be easily mounted and easily adjusted to certain viewing configuration. Many PTZ cameras can output angle (rotation) and zooming factor (focal-length) measurements through an I/O port. However, the accuracy of these measurements is far from adequate for most of the intelligent surveillance applications. Therefore an accurate camera calibration procedure is particularly needed.

Traditional camera calibration is a tedious and laborious procedure, involving the use of a special 3D calibration grid or 2D calibration pattern. It is often conducted off-line (before real camera work). This is obviously not



Figure 1. A typical surveillance PTZ camera system setup.

suitable for a PTZ camera for which rotation and zooming are performed frequently during working. Moreover, for most affordable PTZ CCD cameras, lens zooming often cause some changes in other camera intrinsics (such as the position of the optical center). Therefore, an on-line *auto-calibration* capability is necessary.

In theory, under certain conditions, both camera intrinsic and extrinsic parameters can be estimated by using a moving camera observing a static scene—so called camera *self-calibration* (or *auto-calibration*). The theory of self-calibration was established more than a decade ago. Despite of its theoretical elegance, existing self-calibration algorithms however, have *not* been routinely used in vision systems. A common perception as well as a major criticism of them is the lack of *numerical stability*. For example, sometimes a conventional self-calibration algorithm may end up with a meaningless complex-valued focal-length.

In this paper, we provide a more reliable PTZ camera self-calibration algorithm. The main contribution is a natural way of enforcing certain positive-definiteness constraint which is often required by camera calibration problem, but has seldom been addressed by conventional algorithms. More specifically, we have *formulated* the self-calibration problem into a Linear Matrix Inequality (LMI) optimization form (a special case of *convex programming*), for which a certified and highly efficient numerical algorithm exists. Our experiments on both simulated data and real images have obtained good results.

2. Rotation based camera self-calibration

We use the following notations in this paper. Let a PTZ camera observing a rigid scene. Denote the obtained images as frame-1, frame-2, etc.

Let $K_i = \begin{bmatrix} f & s & x_0 \\ 0 & \alpha f & y_0 \\ 0 & 0 & 1 \end{bmatrix}_i$ be the camera intrinsic matrix

at frame- i , which contains its focal length f , aspect ratio α , skewness s and optical centre x_0, y_0 . In most general case, all these intrinsic parameters are allowed to change.

Define DIAC (dual image of absolute conic) matrix Ω_i as

$$\Omega_i = (K_i K_i^T)^{-1} \quad (1)$$

From this definition it is obvious that Ω_i is *positive definite*, i.e., $\Omega_i \succ 0$. By definition, a symmetric matrix A is called *positive definite* if and only if $\forall x \neq 0, x \in \mathbb{R}^n$ we always have $x^T A x > 0$. A positive definite matrix can be factorized (using Cholesky decomposition) into the form of $B^T B$ where B is an upper triangular matrix (compared with eq.1).

The *significance* of emphasizing such positive-definiteness property lies in the fact that such property is a necessary condition that every valid DIAC matrix Ω_i must hold. In practice, it implies some physical constraints of the camera intrinsics. For example, the focal-length must have real positive values, and the aspect-ratio must be positive real too.

However, this condition has seldom been explicitly enforced in most conventional self-calibration algorithms. There was actually no simple way to do so without much complicating the algorithm (we will explain this shortly).

As a result, using these conventional algorithm one would *very often* obtain a non-positive-definite matrix, from which no valid camera intrinsic matrix can be computed. For example, factorizing a non-positive-definite DIAC matrix one may get a camera intrinsic matrix with complex-valued focal-length. Fig-2 and fig-3 left sides show the success rate (1-failure rate) of conventional algorithms.

2.1. Basic equation for rotation calibration

When a camera is undergoing a pure rotation and capturing images of a static scene, it is possible to recover all the camera intrinsic parameters using the multiple images. This is the basic idea of pure rotation-based self-calibration originally proposed by Hartley [3]. While it might be counter-intuitive, later many experiments have shown that even when the camera motions are not pure rotations but also small translations, the same algorithm still works well and outputs reasonably accurate result. In other words, the algorithm is not sensitive to small translation. Paper [6] and [14] have provided thorough accounts to this property. This is encouraging because such kind of camera operation fits

very well in to a PTZ camera. From now on, we assume that the PTZ camera is purely rotating (and possibly zooming).

Under such pure-rotation assumption, any two images (of a static scene) obtained a PTZ camera are related by a (planar) homography. This is a pure geometric relationship independent of scene contents. We denote such a homography between frame- i and frame- j as H_{ij} . To ease the symbol, let us define its transposed inverse version as $G_{ij} = H_{ij}^{-T}$. The basic equation for pure rotation self-calibration is

$$\Omega_i = G_{ij} \Omega_j G_{ij}^T \quad (2)$$

This is actually a reduced form of the Kruppa equation.

Note that this is in fact an *equation* (rather than a *homogeneous relationship*) if G is normalized so that $\det G = 1$ (c.f [5]).

2.2 Conventional algorithms

There are many variants of rotation-based self-calibration algorithms. Most of them can eventually be reduced to a linear null-space problem: $A c = \mathbf{0}$, where c is the vectorized Ω , and A a coefficient matrix. Therefore, c is simply the null-vector of A . Once c is computed, rearrange it into matrix form of Ω and apply the Cholesky decomposition, one then recovers the camera intrinsic matrix K .

It is noticed that, during this process no special attention has been given to the positive-definiteness of Ω_i . It is not easy to enforce such constraint into the above null-vector computation. As a result, if the computed Ω_i is not positive-definite (which frequently happens), then it is not possible to recover a valid camera intrinsic matrix K using the Cholesky decomposition.

If this happens, one could think of re-projecting the resultant negative semi-definite Ω onto its nearest positive-definite matrix $\hat{\Omega}$ —a trick that is often adopted to enforce rank-deficient constraint in vision computation [5]. However, such post-correction operation often yields spurious solution which may lie on the boundary of the condition and which in general isn't the right solution of the problem. Some researchers suggest using more frames (than necessary) to alleviate the problem. But this strategy does not always work either.

Therefore, a more appropriate approach is to incorporate the positive-definite constraint naturally early in the computation, instead of the post-correction approach. Theoretically it is also possible to formulate the problem as a constrained minimization problem. However, this approach is not recommended either, because it often involves solving a constrained nonlinear optimization problem whose convergence is not always guaranteed in general.

The contribution of the present paper is: we have found that the above basic equation, under the positive-definite constraint, can be naturally cast as an LMI-convex programming problem, for which the positive-definiteness condition can be easily handled and the computation is relatively cheap.

3. LMI convex programming

A linear matrix inequality (LMI) is any constraint of the form

$$F(\mathbf{x}) = F_0 + \sum_{i=1}^N x_i F_i \succ 0;$$

where $\mathbf{x} = (x_1, \dots, x_N)$ is a vector of unknown scalars (called the *decision variables*), F_0, \dots, F_N are given symmetric matrices, and “ $\succ 0$ ” stands for “positive definite”, i.e., the minimal eigenvalue of $F(\mathbf{x})$ is positive. This form is known as the *conic representation* of LMI, namely, affine combination of symmetric positive definite matrices using decision variables.

LMI optimization is a special case of *convex programming*, which is written as

$$\begin{aligned} \min_{\mathbf{x}} \quad & c^T \mathbf{x} \\ \text{such that} \quad & F(\mathbf{x}) = F_0 + \sum_{i=1}^N x_i F_i \succ 0; \end{aligned}$$

For solving this kind of problem, there are extremely efficient (polynomial-time) algorithms as well as off-the-shelf software packages available. SeDuMi [11] is one of the most popular choices of software packages. It solves the following convex programming:

$$\begin{aligned} \min_{\mathbf{x}} \quad & c^T \mathbf{x} \\ \text{such that} \quad & A\mathbf{x} = b, \mathbf{x} \in D \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $\mathbf{x}, c \in \mathbb{R}^n$, D is a cone (e.g., an LMI form).

4. The case of self-calibrating a PT camera

We consider two typical cases of PTZ camera usage, which are PT case and PTZ case. In the PT case, the camera is only allowed to pan and tilt. Therefore, the camera is only undergoing a two-axis rotation, but no zooming is allowed. For such case, it is quite safe to assume the camera intrinsic matrix is constant over frames.

Using the constant intrinsic assumption, one can fulfil a self-calibration by solving the following basic equation (note that the subscription of Ω has been removed due to the constant intrinsic assumption):

$$\Omega = G_{ij} \Omega G_{ij}^T. \quad (3)$$

In what follows, we will describe how to transform this equation into an LMI form, based on which the positive-definite condition can be naturally embedded into the computation. Without loss of generality, we describe our algorithms based on three frames, although the same algorithm can be trivially extended to arbitrarily many views.

Suppose three views, say view 1, 2 and 3, are given. $G_{i,j}$ denotes the inverse transposed homography from view i to view j . Ω is the constant DIAC to be estimated. Recall that a valid DIAC must be positive definite. In the first step, we reformulate eq.(3) as a constrained minimization problem:

$$\min_{\Omega} \{ \|\Omega - G_{12} \Omega G_{12}^T\|_2 + \|\Omega - G_{13} \Omega G_{13}^T\|_2 \} \quad (4)$$

$$\text{such that} \quad \Omega \succ 0 \quad (5)$$

where $\|\cdot\|_2$ represents the L_2 norm.

Next, introduce two upper bound variables $t_1 \geq 0$ and $t_2 \geq 0$. Then the problem is equivalent to

$$\min_{\Omega} (t_1 + t_2) \quad (6)$$

$$\text{such that} \quad \|\Omega - G_{12} \Omega G_{12}^T\|_2 < t_1 \quad (7)$$

$$\|\Omega - G_{13} \Omega G_{13}^T\|_2 < t_2 \quad (8)$$

$$\Omega \succ 0 \quad (9)$$

Now we write the three conditions (inequalities and positive definite) into matrix form. They actually are

Condition-1:

$$C_1 = \begin{bmatrix} t_1 I & \Omega - G_{12} \Omega G_{12}^T \\ (\Omega - G_{12} \Omega G_{12}^T)^T & t_1 I \end{bmatrix} \succ 0 \quad (10)$$

Condition-2:

$$C_2 = \begin{bmatrix} t_2 I & \Omega - G_{13} \Omega G_{13}^T \\ (\Omega - G_{13} \Omega G_{13}^T)^T & t_2 I \end{bmatrix} \succ 0 \quad (11)$$

Condition-3:

$$C_3 = \Omega \succ 0 \quad (12)$$

It is easily seen that the three matrix-form conditions are equivalent to eqns.(7)(8)(9) respectively, by checking the determinants of the three matrices.

Absorbing these three matrices into a big block matrix, we then obtain the following optimization problem, which

is equivalent to the original Eq.(4).

$$\begin{aligned} \min_{\Omega} \quad & (t_1 + t_2) \\ \text{such that} \quad & \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \succ 0 \end{aligned} \quad (13)$$

So far we have converted the original equation (3) into a convex programming problem (13). In order to use the SeDuMi, we need to further translate it into a conic LMI form.

Let us firstly show how condition-3 (viz. a single positive definite condition) can be translated into an LMI form:

Let $\Omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \\ \omega_2 & \omega_4 & \omega_5 \\ \omega_3 & \omega_5 & \omega_6 \end{bmatrix}$. Let $P_i, i = 1 \dots 6$ be the basis matrices for a 3×3 symmetric matrix, i.e., $P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, P_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \dots$ It is easy to reach:

$$\Omega = \sum_{i=1}^6 \omega_i P_i \quad (14)$$

Secondly, we will show that **condition-1** and **-2** are both conic LMI forms too.

Substituting Eq.(14) into condition-1 (or -2), we obtain:

$$\begin{aligned} C_1 &= \begin{bmatrix} t_1 \mathbf{I} & \Omega - G_{12} \Omega G_{12}^T \\ (\Omega - G_{12} \Omega G_{12}^T)^T & t_1 \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} t_1 \mathbf{I} & \sum_{i=1}^6 \omega_i (P_i - G_{12} P_i G_{12}^T) \\ \sum_{i=1}^6 \omega_i (P_i - G_{12} P_i G_{12}^T)^T & t_1 \mathbf{I} \end{bmatrix} \\ &= t_1 \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{bmatrix} + \sum_{i=1}^6 \omega_i \begin{bmatrix} 0 & P_i - G_{12} P_i G_{12}^T \\ (P_i - G_{12} P_i G_{12}^T)^T & 0 \end{bmatrix} \\ &= t_1 \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{bmatrix} + \sum_{i=1}^6 \omega_i F_i \succ 0 \end{aligned}$$

Now define the vector of decision variables as

$$\mathbf{x} = [t_1, t_2, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6] \in \mathbb{R}^8.$$

We now have reached a big LMI problem which is formulated as

$$\min_{\mathbf{x}} \quad (t_1 + t_2) \quad (15)$$

$$\text{such that} \quad t_1 \mathcal{I} + t_2 \mathcal{I} + \sum_{i=1}^6 \omega_i \mathcal{F}_i \succ 0, \quad (16)$$

where \mathcal{F} and \mathcal{I} are appropriate symmetric matrices. This is a standard form of convex LMI optimization problem. Such completes our derivation, and the SeDuMi function is readily usable.

5. The case of self-calibrating a PTZ camera

If the camera allows to zoom freely, then the constant intrinsic assumption is no longer hold. This is not only because the focal-length (i.e. zooming factor) is an element of the intrinsic parameter matrix, but because for most commonly used PTZ cameras the change of focal-length often cause other intrinsic parameters change as well. This is partially due to the inaccuracy in camera mechanical and optical design. Typically, zooming a lens often makes the optical centre shift. Consequently, for a PTZ camera, one generally can not assume a fixed intrinsic matrix. Rather, other knowledge about the camera must be exploited.

It is quite convenient and safe to assume the camera has rectangle pixels, in other words, the skewness is zero. However, using zero-skewness only is not enough to self-calibrate a PTZ camera. Because the rotation of a PTZ camera is not a full 3-axis rotation, but only a 2-axis. This situation actually corresponds to a so-called *degenerate* case for camera self-calibration. When camera undergoes a degenerate motion, there will be inherent ambiguities in the calibration algorithm. In general, one would not get any meaningful result. For example, if the camera is purely panning, then the focal-length in y-axis f_y is un-recoverable.

To solve this problem, one must add other constraint. Here, we further assume the aspect ratio is known. This assumption is not too restrictive in practice, because the aspect ratio is generally fixed during the camera zooming and rotating. Therefore, one only needs to estimate it once and uses it afterward.

Now we are in the position of explaining how to simultaneously enforce the zero-skewness and unit-aspect ratio constraints in the form of LMI optimization. Without loss of the generality, let us assume the aspect ratio is unit.

5.1. Enforce $s = 0$ and $\alpha = 1$

Suppose camera intrinsic matrix at frame i is Ω_i . The two constraints are thus:

$$(\Omega_i)_{1,2} = 0 \quad (17)$$

and

$$(\Omega_i)_{1,1} = (\Omega_i)_{2,2}$$

where $(1, 2), (1, 1), (2, 2)$ stands for the $(1,2), (1,1), (2,2)$ elements. Since SeDuMi accepts matrix form inputs only, we introduce two vectors $\mathbf{L} = [1, 0, 0]^T$ and $\mathbf{R} = [0, 1, 0]^T$. The above two equations become

$$\mathbf{L}^T \Omega_i \mathbf{R} = 0 \quad (18)$$

and

$$\mathbf{L}^T \Omega_i \mathbf{L} = \mathbf{R}^T \Omega_i \mathbf{R} = 0$$

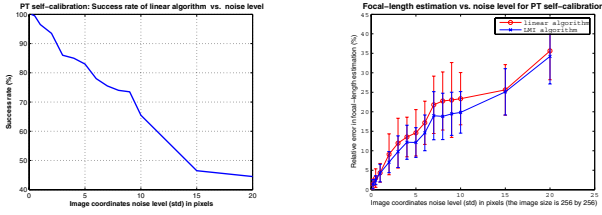


Figure 2. Self-calibration of a PT camera. (left: success rate of a linear algorithm vs noise; right: error in estimation)

Using Eq.(2), then we have

$$\mathbf{L}^T \mathbf{G}_{ij} \Omega_i \mathbf{G}_{ij}^T \mathbf{R} = 0 \quad (19)$$

$$\mathbf{L}^T \mathbf{G}_{ij} \Omega_i \mathbf{G}_{ij}^T \mathbf{L} = \mathbf{R}^T \mathbf{G}_{ij} \Omega_i \mathbf{G}_{ij}^T \mathbf{R} = 0 \quad (20)$$

Now each input frame j provides two equations on the DIAC matrix Ω_i . As Ω_i has 3 DOFs only (f, x_0, y_0), in theory two frames are sufficient to obtain an estimation of Ω .

Because there are inevitable noise in the estimated homographies (and in \mathbf{G}_{ij}), to alleviate it, in practice one often needs more than two frames and often an optimization scheme is adopted for the purpose of better robustness.

We now show that by formulating the problem as optimization will naturally give rise to a convex LMI form. Introducing upper bound variables $t_i \geq 0$ for each frame i and following a similar derivation of the previous section, we have obtained two LMI conditions shown below:

$$\begin{bmatrix} t_i & \mathbf{L}^T \mathbf{G}_{ij} \Omega_i \mathbf{G}_{ij}^T \mathbf{R} \\ (\mathbf{L}^T \mathbf{G}_{ij} \Omega_i \mathbf{G}_{ij}^T \mathbf{R})^T & t_i \end{bmatrix} \succ 0 \quad (21)$$

$$\text{and } \begin{bmatrix} t_i & \mathbf{M} \\ \mathbf{M}^T & t_i \end{bmatrix} \succ 0, \quad (22)$$

where $\mathbf{M} = \mathbf{L}^T \mathbf{G}_{ij} \Omega_i \mathbf{G}_{ij}^T \mathbf{L} - \mathbf{R}^T \mathbf{G}_{ij} \Omega_i \mathbf{G}_{ij}^T \mathbf{R}$. Solving the LMI problem of $\min \sum_i t_i$ under the above two LMI conditions concludes this section. It is worth noting that here we use different bound variables t_i for different frame. Alternatively, one can share a common upper bound t . This actually corresponds to an L_∞ optimization strategy [4], and it will not affect the direction of minimization. To strictly guarantee the two constraints for each frame, our implementation also uses an idea of sub-space expansion (c.f. [5] page-594, Algorithm A5.5).

6. Calibrating extrinsics: motion estimation

Once the intrinsic matrices at each frame are obtained, one can easily compute the camera extrinsic parameters. For a PTZ camera, these extrinsic parameters are no other than the two rotations—pan and tilt. As we argued in the

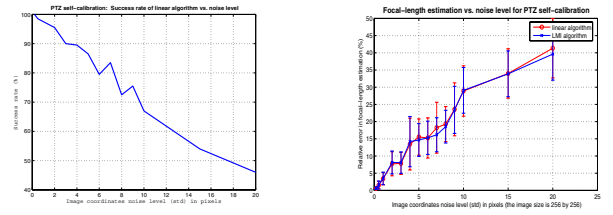


Figure 3. Self-calibration of a PTZ camera. (left: success rate of a linear algorithm vs noise; right: error in estimation)

first section, because they are actually measured from the input image signal, they are more suitable for the task of intelligent visual surveillance.

One simple way to compute the rotation \mathbf{R}_i between frame i and j is through

$$\mathbf{R}_i = \mathbf{K}_i^{-1} \mathbf{H}_{ij} \mathbf{K}_j^{-T}.$$

If translation estimation is also required (although it is small), one can use the classic eight-point algorithm, or our recently improved five-point algorithm [8].

7. Experiments

We simulate 100 3D points randomly distributed in a cube. Then project them into images through a perspective camera. The camera is undergoing pure rotation and zooming. The simulated image size is about 256 by 256. Gaussian noise of different levels is introduced to image coordinates. The homography between each frame-pair is estimated by a simple normalized DLT linear algorithm. No non-linear iterative refinement is used. For both the PT case and the PTZ case we estimate the camera intrinsic matrices using our LMI algorithms, and compare them with the results obtained by conventional linear algorithms, as well as with the ground-truth. In experiments, we evaluate both the success rate—by which we merely mean that the obtained Ω is indeed positive definite, and the accuracy of estimation in term of the value of focal-length.

7.1. PT camera self-calibration

In the PT experiment, we used 3 frames and assume constant intrinsics. All tests are repeated 1000 times and the average results are shown in figure-2. The left side gives the success rate of linear algorithm. It is seen that as the noise level increases the success rate of the linear algorithm falls quickly, while the new LMI algorithm always gives 100% success rate. For those cases when the linear algorithm is



Figure 4. Sample frames of a hand-held panning sequence.

successful, we compare the estimation results with those by the LMI algorithm. Figure-2 right gives the relative error (percentage) in focal-length estimation. It is seen that they are very close.

7.2. PTZ camera self-calibration

We also conduct similar experiments for the PTZ case. The following figure (fig-3) shows the success rate and estimation accuracy w.r.t. noise levels. Similar conclusions can be drawn that: with guaranteed 100% success rate our LMI algorithm also achieves accurate calibration. In term of computation time, both linear algorithm and our LMI algorithm (based on SeDuMi) are fast enough for on-line calibration, often below 0.01 second on a moderate PC (Intel P-4 CPU@2Ghz, 1G memory) using MATLAB.

7.3. Test on real images:hand-held camera

We use a hand-held video camcorder performing panning motion roughly about a fix point. The focal length is fixed during the imaging process. We take 20 frames of a static scene (some samples are shown in figure-4). A KLT tracker ([9]) is applied to get point correspondences. Using our LMI PTZ self-calibration algorithm (without assuming constant intrinsic) we obtain constant focal-length estimations. We get 100% successful results for all 20 frames. As the result is simply a horizontal line (i.e., 20 constant focal-lengths), we omit its figure here also due to space limit. This has validated our algorithm.

8 Closing remarks

This paper has shown how to re-formulate the rotation-based camera self-calibration as an LMI convex optimization problem. By such, the positive-definiteness constraint has been enforced naturally, and the resultant new algorithm has achieved more reliable calibration. We apply the new algorithm to PTZ camera self-calibration, and obtain satisfactory result.

Besides the above advantages, using such LMI formulation has also other benefits. Thanks to the simplicity of the LMI form, various priors about the camera can also be incorporated in a natural way. For example, knowledge about the bounds of some of the intrinsic camera parameters can be easily posed as new LMI inequality conditions. Moreover, some rough measurements of camera rotation (e.g, from mechanical sensors) can be converted to LMI norm constraint on the camera calibration. These will be our future research.

Based on the results, we conclude that the pure-rotation self-calibration problem is essentially a simple convex LMI problem. That is the reason why the final LMI forms are so naturally obtained. In computer vision, there are also many non-convex problems. For some of those problems, by certain relaxation approximation, this LMI approach could still be applicable, see for example Kahl-2005 [7]. In short, the LMI optimization formulation is a promising and principled approach of incorporating user prior knowledge, and could be very useful in practice.

Acknowledgement. The authors wish to thank anonymous reviewers' invaluable comments. Thank the second reviewer who suggested explaining more on the practical implication of the positive-definite condition, and removing confusing short-hands. All these largely improve the paper presentation. NICTA is funded through the Australian Government's Backing Australia's Ability Initiative, in part through the ARC.

References

- [1] L. Agapito, R. Hartley, and E. Hayman. Linear calibration of a rotating and zooming camera. *CVPR*, pages 15–21, 1999.
- [2] M. Agrawal. On automatic determination of varying focal lengths using semidefinite programming. *ICIP*, October 2004.
- [3] R. Hartley. Self-calibration from multiple views with a rotating camera. *Proc. ECCV*, pages 471–478, 1994.
- [4] R. Hartley and F. Schaffalitzky. L-infinity minimization in geometric reconstruction problems. *CVPR*, 01:504–509, 2004.
- [5] R. Hartley and A. Zisserman. *Multiview Geometry in computer vision*. Cambridge University Press, 2004.
- [6] E. Hayman and D. Murray. The effects of translational misalignment when self-calibrating rotating and zooming cameras. *IEEE T-PAMI*, 25:1015–1020, 2003.
- [7] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. *Proc ICCV-2005*, 2:978–985, 2005.
- [8] H. Li and R. Hartley. Five-point algorithm made easy. *Proc. ICPR*, 1, 2006.
- [9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, pages 674–679, 1981.
- [10] S. Sinha and M. Pollefeys. Towards calibrating a pan-tilt-zoom camera network. *OMNIVIS*, 2004.
- [11] J. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, pages 625–653, 1999.
- [12] M. Trajkovic. Interactive calibration of a ptz camera for surveillance applications. *ACCV*, 2002.
- [13] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.
- [14] L. Wang and S. B. Kang. Error analysis of pure rotation-based self-calibration. *IEEE T-PAMI*, 26(2):275–280, 2004.