UC San Diego
UC San Diego Previously Published Works

Title
Globally Optimal Algorithms for Stratified Autocalibration

Permalink
https://escholarship.org/uc/item/5fw7c712

Journal
International Journal of Computer Vision, 90(2)

ISSN
1573-1405

Authors
Chandraker, Manmohan
Agarwal, Sameer
Kriegman, David
et al.

Publication Date
2010-11-01

DOI
10.1007/s11263-009-0305-2

Peer reviewed

Globally Optimal Algorithms for Stratified Autocalibration

**Manmohan Chandraker · Sameer Agarwal ·
David Kriegman · Serge Belongie**

Abstract We present practical algorithms for stratified autocalibration with theoretical guarantees of global optimality. Given a projective reconstruction, we first upgrade it to affine by estimating the position of the plane at infinity. The plane at infinity is computed by globally minimizing a least squares formulation of the modulus constraints. In the second stage, this affine reconstruction is upgraded to a metric one by globally minimizing the infinite homography relation to compute the dual image of the absolute conic (DIAC). The positive semidefiniteness of the DIAC is explicitly enforced as part of the optimization process, rather than as a post-processing step.

For each stage, we construct and minimize tight convex relaxations of the highly non-convex objective functions in a branch and bound optimization framework. We exploit the inherent problem structure to restrict the search space for the DIAC and the plane at infinity to a small, fixed number of branching dimensions, independent of the number of views. Chirality constraints are incorporated into our convex relaxations to automatically select an initial region which is guaranteed to contain the global minimum.

M. Chandraker (✉) · D. Kriegman · S. Belongie
Department of Computer Science and Engineering, University of California, San Diego, USA
e-mail: mkchandraker@cs.ucsd.edu

D. Kriegman
e-mail: kriegman@cs.ucsd.edu

S. Belongie
e-mail: sjb@cs.ucsd.edu

S. Agarwal
Department of Computer Science and Engineering, University of Washington, Seattle, USA
e-mail: sagarwal@cs.washington.edu

Experimental evidence of the accuracy, speed and scalability of our algorithm is presented on synthetic and real data.

1 Introduction

This paper proposes practical algorithms that provably solve well-known formulations for both affine and metric upgrade stages of stratified autocalibration to their global optimum.

Given n feature correspondences across n views of a scene, it is well-known that a projective reconstruction may be computed that differs from the true scene by an arbitrary 4×4 linear transformation, or homography (Faugeras 1992; Hartley et al. 1992). A projective reconstruction may be upgraded to a metric one, which differs from the true scene by a similarity transformation, using *a priori* knowledge of a few scene characteristics, such as the angles between a few 3D lines. An alternative approach to estimate the 4×4 transformation that restores the metric scene is through simple assumptions on the internal parameters of the cameras used to image the scene, such as their constancy across different views, or the rectangularity of the image pixels. The latter approach is called autocalibration, or camera self-calibration, which forms the subject of this paper.

A typical approach to calibrating a camera involves using several images of a known calibration grid. Once a correspondence can be ascertained between scene points (or higher order features like curves) and their counterparts on the image plane, it is straightforward to recover the camera parameters. The term autocalibration stems from its key

premise that it obviates the requirement for an explicit calibration grid. Instead, it tries to locate the image of the so-called absolute conic, which is an imaginary object on the plane at infinity, whose location stays fixed in any metric reconstruction. Its image can be shown to be related to the internal parameters of the camera, so locating the image of the absolute conic is equivalent to calibrating the camera.

The method of autocalibration presented in this paper is a stratified one, whose first step upgrades the projective reconstruction to an affine one, while the next step performs the upgrade to a metric reconstruction. An affine reconstruction restores certain aspects of the scene, such as parallelism between 3D lines, while the metric reconstruction restores characteristics such as exact angles and length ratios.

The affine upgrade, which is arguably the more difficult step in stratified autocalibration, is succinctly computable by estimating the position of the plane at infinity in a projective reconstruction, for instance, by solving the modulus constraints (Pollefeys and Gool 1999). Previous approaches to minimizing the modulus constraints for several views rely on local, gradient-based methods with random reinitializations. These methods are not guaranteed to perform well for such non-convex problems. Moreover, in our experience, a highly accurate estimate of the plane at infinity is imperative to obtain a usable metric reconstruction.

The metric upgrade step involves estimating the intrinsic parameters of the cameras, which is commonly approached by estimating the dual image of the absolute conic (DIAC). A variety of linear methods exist towards this end, however, they are known to perform poorly in the presence of noise (Hartley and Zisserman 2004). Perhaps more significantly, most methods *a posteriori* impose the positive semi-definiteness of the DIAC, which might lead to a spurious calibration. Thus, it is important to impose the positive semi-definiteness of the DIAC *within* the optimization, not as a post-processing step.

This paper proposes global minimization algorithms for both stages of stratified autocalibration that furnish theoretical certificates of optimality. That is, they return a solution at most ϵ away from the global minimum, for arbitrarily small ϵ. Our solution approach relies on constructing efficiently minimizable, tight convex relaxations to non-convex programs and using them in a branch and bound framework (Horst and Tuy 2006; Tawarmalani and Sahinidis 2002). A preliminary version of this paper appeared in (Chandraker et al. 2007b).

A significant drawback of local methods is that they critically depend on the quality of a heuristic initialization. To be considered truly optimal, an algorithm must converge to the global optimum regardless of the choice of initialization. Branch and bound methods require a demarcated region of the search space as initialization. Arbitrarily choosing a small initial region might compromise optimality, since the true solution might lie outside that chosen region. On the other hand, choosing a very large region might lead to a ponderous convergence rate for the branch and bound algorithm.

On the other hand, in this paper, we use chirality constraints derived from the scene to compute a theoretically correct initial search space for the plane at infinity, within which we are guaranteed to find the global minimum (Hartley 1998; Hartley et al. 1999). In practice, for a moderate number of cameras, the initial region determined by the chirality constraints are tight enough to allow rapid convergence of the search algorithm. Our initial region for the metric upgrade is intuitively specifiable as conditions on the intrinsic parameters of the camera and can be wide enough to include any practical case.

A crucial concern in branch and bound algorithms is the exponential dependence of the worst case time complexity on the number of branching dimensions. The number of branching dimensions in most computer vision problems scales with the number of points and views, which can quickly translate into an impractical branch and bound search. In this paper, we exploit the inherent problem structure of autocalibration to restrict our branching dimensions to a small, fixed number, independent of the number of views. In our experiments, this allows the runtime of algorithms proposed in this paper to scale gracefully with the number of views.

In summary, our main contributions are the following:

– Highly accurate recovery of the plane at infinity in a projective reconstruction by global minimization of the modulus constraints.
– Highly accurate estimation of the DIAC by globally solving the infinite homography relation.
– A general exposition on novel convexification methods for global optimization of non-convex programs.

The outline of the rest of the paper is as follows. Section 2 describes background relevant to autocalibration and Sect. 3 outlines the related prior work. Section 4 is a brief overview of branch and bound algorithms. Section 5 describes the general strategy that we employ for constructing the convex relaxation of a non-convex function, while Sects. 6 and 7 describe our global optimization algorithms for estimating the plane at infinity and the DIAC, respectively. Section 8 presents experiments on synthetic and real data and Sect. 9 concludes with a discussion of further extensions.

2 Background

Unless stated otherwise, we will denote 3D world points $\mathbf{X}$ by homogeneous 4-vectors and 2D image points $\mathbf{x}$ by homogeneous 3-vectors. Let $\widehat{P}_i$, $i = 1, \ldots, m$, be Euclidean cameras parameterized as 3×4 matrices, that observe a scene

composed of 3D points $\widehat{bX}_j$, $i = 1, \ldots, n$, resulting in image points $\mathbf{x}_{ij}$ consistent with perspective projection:

$$x_{ij} = \widehat{P}_i \widehat{\mathbf{X}}_j, \quad i = 1, \ldots, m, \; j = 1, \ldots, n, \tag{1}$$

where the above equality holds up to a scale factor. Then, a projective reconstruction can be computed as a set of cameras P_i, $i = 1, \ldots, m$ and a set of points $\mathbf{X}_j$, $j = 1, \ldots, n$ satisfying the following equalities up to a scale factor:

$$x_{ij} = P_i \mathbf{X}_j, \quad i = 1, \ldots, m, \; j = 1, \ldots, n. \tag{2}$$

A factorization-based approach that extends the affine factorization method of (Tomasi and Kanade 1992) to the projective case is presented in (Sturm and Triggs 1996).

From (1) and (2), it is apparent that the projective reconstruction is related to the Euclidean one by an arbitrary 4×4 homography:

$$\begin{aligned} P_i &= \widehat{P}_i H^{-1}, \quad i = 1, \ldots, m, \\ \mathbf{X}_j &= H \widehat{\mathbf{X}}_j, \quad j = 1, \ldots, n. \end{aligned} \tag{3}$$

Given a projective reconstruction, autocalibration seeks to estimate the best homography H that upgrades the reconstruction to a metric one.

A more detailed discussion of the material in this section can be found in (Hartley and Zisserman 2004).

2.1 The Infinite Homography Relation

The Euclidean camera is parametrized as $\widehat{P} = K[R|\mathbf{t}]$ where the 3×3 rotation matrix R and the 3×1 translation vector $\mathbf{t}$ constitute the exterior orientation, while the 3×3 upper-triangular matrix K encodes the intrinsic parameters of the camera.

We can always perform the projective reconstruction such that $P_1 = [I|\mathbf{0}]$. Let the world coordinate system be aligned with the first (world) camera, i.e. $\widehat{P}_1 = K_1[I|\mathbf{0}]$. Then, it can be easily shown that H has the form

$$H = \begin{bmatrix} K_1 & \mathbf{0} \\ -\mathbf{p}^\top K_1 & 1 \end{bmatrix}, \tag{4}$$

where $\pi_\infty = (\mathbf{p}, 1)^\top$ is the location to which the plane at infinity moves in the projective reconstruction from its canonical position $(0, 0, 0, 1)^\top$. Thus, the aim of autocalibration is to recover the plane at infinity and the intrinsic parameters.

Let $P_i = [A_i|\mathbf{a}_i]$ where A_i is a 3×3 matrix and $\mathbf{a}_i$ is a 3×1 vector. Let $\widehat{P}_i = K_i[R_i|\mathbf{t}_i]$. Then, assuming that the internal parameters of the camera remain the same across all the views, that is, $K_i = K$, we have

$$\omega^* = (A_i - \mathbf{a}_i \mathbf{p}^\top) \omega^* (A_i - \mathbf{a}_i \mathbf{p}^\top)^\top, \tag{5}$$

where $\omega^* = KK^\top$ is a 3×3 homogeneous, symmetric matrix that denotes the dual image of the absolute conic (DIAC).

Note that equality in (5) is up to a scale factor. Since $H_\infty^i = (A_i - \mathbf{a}_i \mathbf{p}^\top)$ is precisely the homography induced between the views $P_1 = [I|0]$ and $P_i = [A_i|\mathbf{a}_i]$ by the plane at infinity $(\mathbf{p}, 1)^\top$, it is called the infinite homography (Hartley and Zisserman 2004).

2.2 Modulus Constraints

It follows from (3) and (4) that, for the case of constant intrinsic parameters,

$$KR = (A_i - \mathbf{a}_i \mathbf{p}^\top)K.$$

Thus, noting that the infinite homography is conjugate to a rotation matrix and must have eigenvalues of equal moduli, one can relate the roots of its characteristic polynomial

$$\det(\lambda I - (A_i - \mathbf{a}_i \mathbf{p}^\top)) = \lambda^3 - \alpha_i \lambda^2 + \beta_i \lambda - \gamma_i \tag{6}$$

to derive the so called *modulus constraint* (Pollefeys and Gool 1999):

$$\gamma_i \alpha_i^3 = \beta_i^3, \tag{7}$$

where $\alpha_i, \beta_i, \gamma_i$ are affine functions of the coordinates $\{p_1, p_2, p_3\}$ of the plane at infinity. Three views suffice to restrict the solution space to a $4^3 = 64$ possibilities (actually 21, see Schaffalitzky 2000), which can in theory be extracted using continuation methods.

2.3 Chirality Bounds on Plane at Infinity

Chirality constraints demand that the reconstructed scene points lie in front of the camera. While a general projective transformation may result in the plane at infinity splitting the scene, a quasi-affine transformation is one that preserves the convex hull of the scene points $\mathbf{X}$ and camera centers $\mathbf{C}$. A transformation H_q that upgrades a projective reconstruction to quasi-affine can be computed by solving the so-called chiral inequalities. A subsequent affine centering, H_a, guarantees that the plane at infinity in the centered quasi-affine frame, $\mathbf{v} = (H_a H_q)^{-\top} \pi_\infty$, cannot pass through the origin. So it can be parametrized as $(v_1, v_2, v_3, 1)^\top$ and bounds on v_i in the centered quasi-affine frame can be computed by solving six linear programs:

$$\left. \begin{aligned} &\min / \max \; v_i \\ &\text{subject to } \mathbf{X}_j^{q\top} \mathbf{v} > 0, \quad j = 1, \ldots, n \\ &\qquad\qquad \mathbf{C}_k^{q\top} \mathbf{v} > 0, \quad k = 1, \ldots, m \end{aligned} \right\} \quad i = 1, 2, 3, \tag{8}$$

where $\mathbf{X}_j^q$ and $\mathbf{C}_k^q$ are points and camera centers in the quasi-affine frame. We refer the reader to (Hartley 1998; Nistér 2004) for a thorough treatment of the subject.

3 Previous Work

Approaches to autocalibration (Faugeras et al. 1992) can be broadly classified as direct and stratified. Direct methods seek to compute a metric reconstruction by estimating the absolute conic. This is encoded conveniently in the dual quadric formulation of autocalibration (Heyden and Åström 1996; Triggs 1997), whereby an eigenvalue decomposition of the estimated dual quadric yields the homography that relates the projective reconstruction to Euclidean. Linear methods (Pollefeys et al. 1998) as well as more elaborate SQP based optimization approaches (Triggs 1997) have been proposed to estimate the dual quadric, but perform poorly with noisy data. Methods such as (Manning and Dyer 2001) which are based on the Kruppa equations (or the fundamental matrix), are known to suffer from additional ambiguities (Sturm 2000).

This work primarily deals with a stratified approach to autocalibration (Pollefeys and Gool 1999). It is well-established in literature that, in the absence of prior information about the scene, estimating the plane at infinity represents the most significant challenge in autocalibration (Hartley et al. 1999). The modulus constraints (Pollefeys and Gool 1999) are a necessary condition for the coordinates of the plane at infinity. Local techniques are used in (Pollefeys and Gool 1999) to estimate the coordinates of the plane at infinity by minimizing a noisy overdetermined system in the multi-view case.

An alternate approach to estimating the plane at infinity exploits the chirality constraints. The algorithm in (Hartley et al. 1999) computes bounds on the plane at infinity and a brute force search is used to recover π_∞ within this region. It is argued in (Nistér 2004) that it might be advantageous to use camera centers alone when using chirality constraints.

Several linear methods exist for estimating the DIAC (Hartley and Zisserman 2004) for the metric upgrade, but they do not enforce its positive semi-definiteness. The only work the authors are aware of which explicitly deals with this issue is (Agrawal 2004), which is formulated under the assumption of known principal point and zero skew. The interested reader is referred to (Hartley and Zisserman 2004) and the references therein for a more detailed overview of literature relevant to autocalibration.

Of late, there has been significant activity towards developing globally optimal algorithms for various problems in computer vision. The theory of convex linear matrix inequality (LMI) relaxations (Lasserre 2001) is used in (Kahl and Henrion 2005) to find global solutions to several optimization problems in multiview geometry, while (Chandraker et al. 2007a) discusses a direct method for autocalibration using the same techniques. Triangulation and resectioning are solved with a certificate of optimality using convex relaxation techniques for fractional programs

in (Agarwal et al. 2006). Several geometric problems in computer vision, when posed in the L_∞-norm, can be solved to their global optimum using techniques of quasi-convex optimization (Kahl 2005; Sim and Hartley 2006; Agarwal et al. 2008). A survey of some of the recent work in developing optimal algorithms for multiview geometry can be found in (Hartley and Kahl 2007).

An interval analysis based branch and bound method for autocalibration is proposed in (Fusiello et al. 2004), however the fundamental matrix based formulation does not scale well beyond a small number of views. Gröbner basis methods have been used to achieve optimal solutions for several geometric reconstruction problems, such as triangulation (Stewénius et al. 2005), but they do not scale well for more than a very few number of views.

Branch and bound as a solution paradigm has been used for a diverse range of applications in computer vision, such as feature selection (Zongker and Jain 1996), geometric matching (Breuel 2002), image segmentation (Gat 2003), contour tracking (Freedman 2003), object localization (Lampert et al. 2008) and so on. A branch and bound method for Euclidean registration problems is presented in (Olsson et al. 2009).

4 Branch and Bound Theory

Branch and bound algorithms are non-heuristic methods for global optimization of non-convex problems (Land and Doig 1960). They maintain a provable upper and/or lower bound on the (globally) optimal objective value and terminate with a certificate proving that the solution is ϵ-suboptimal (that is, within ϵ of the global optimum), for arbitrarily small ϵ. For greater details on the branch and bound framework for global optimization, we refer the reader to standard texts such a (Horst and Tuy 2006).

Consider a multivariate, non-convex, scalar-valued objective function $f(x)$, for which we seek a global minimum over a rectangle Q_0. Branch and bound algorithms require an auxiliary function $f_{lb}(Q)$ which for every region $Q \subseteq Q_0$, satisfies two properties:

(L1) The value of $f_{lb}(Q)$ is always less than or equal to the minimum value $f_{min}(Q)$ of $f(x)$ for all $x \in Q$.

(L2) Let $|Q|$ denote the size of a rectangle, which in our case, is the length of the longest edge. Then the relaxation gap $f(x) - f_{lb}(x)$ monotonically decreases as a function of $|Q|$.

Note that while (L1) is a basic stipulation for a convex underestimator, (L2) is a Cauchy continuity requirement specific to branch and bound algorithms. Indeed, several popular convex underestimators such as linear matrix inequality (LMI) relaxations (Lasserre 2001) and sum-of-squares relaxations for polynomial systems (Prajna et al. 2002) do not
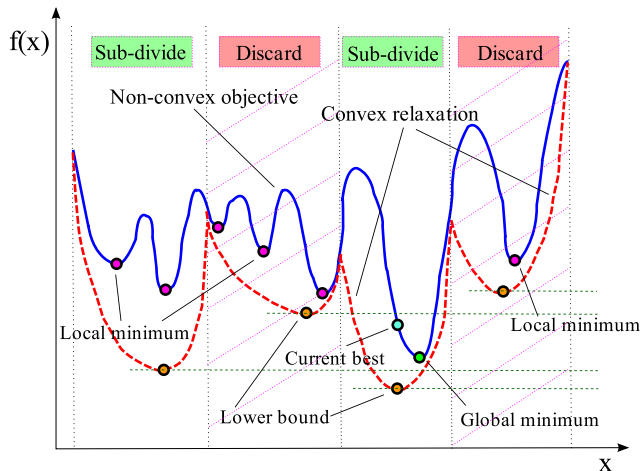
Fig. 1 The basic mechanism of global optimization using a branch and bound framework, illustrated for a univariate function

satisfy this requirement, thus they are rendered unsuitable for our purposes.

Computing the value of $f_{lb}(Q)$ is referred to as *bounding*, while choosing and subdividing a rectangle is called *branching*. The basic mechanism of a branch and bound routine is illustrated for the case of a univariate function in Fig. 1. The choice of the rectangle picked for refinement and the actual subdivision itself are essentially heuristic. We consider the rectangle with the smallest minimum of f_{lb} as the most promising to contain the global minimum and subdivide it into $k = 2$ rectangles along the largest dimension. A key consideration when designing bounding functions is the ease with which they can be estimated. So, it is desirable to design $f_{lb}(Q)$ as the solution of a convex optimization problem for which efficient solvers exist (Boyd and Vandenberghe 2004). In the following sections, we present branch and bound algorithms based on such constructions.

Although guaranteed to find the global optimum (or a point arbitrarily close to it), the worst case complexity of a branch and bound algorithm is exponential. While this may initially appear to be discouraging, we will show in our experiments that exploiting problem structure leads to fast convergence rates in practice.

5 Constructing Convex Relaxations

In this section, we will outline our general strategy for the construction of a convex underestimator for an arbitrary non-convex function. This strategy will be employed to underestimate the objective functions that arise in both the affine and metric upgrade stages of autocalibration.

Let us consider the following unconstrained, non-linear least squares problem:

$$\min_x \sum_{i=1}^{m} (f_i(x) - \mu_i)^2, \tag{9}$$

where $\mu_i \in \mathbb{R}$ and $x \in \mathbb{R}^k$, $k \geq 1$. Then, an equivalent constrained optimization problem is:

$$\min_{x, s_i} \sum_i (s_i - \mu_i)^2 \tag{10}$$

subject to $s_i = f_i(x)$.

Suppose we can construct a convex underestimator $\text{conv}(f_i)$ and a concave overestimator $\text{conc}(f_i)$ for the function $f_i(x)$. Then, the following convex optimization problem minimizes the same objective as (10), but with a "relaxed" constraint set:

$$\min_{x, s_i} \sum_i (s_i - \mu_i)^2 \tag{11}$$

subject to $\text{conv}(f_i) \leq s_i \leq \text{conc}(f_i)$.

Consequently, the minimum attained by the problem (11) will always be at least as low as the minimum attained by (10). In effect, we have constructed a convex problem whose minimum always underestimates the minimum of the non-convex problem we wished to optimize. The solution to (11) corresponds to the construction of the lower bounding function f_{lb} discussed in Sect. 4.

An intuitive illustration of the procedure is depicted for a 1-D function in Fig. 2. While the variable s is allowed to attain values only on the graph of the function $f(x)$ in the original problem (10), it can attain any value within the larger region between the convex and concave relaxations in the relaxed problem (11).

6 Global Estimation of Plane at Infinity

6.1 Traditional Solution

Given exactly three views, the modulus constraints of (7) correspond to a system of three quartic polynomials in three variables, for which the 64 roots may be found, typically using continuation methods. Also for the three-view case, an additional cubic equation available from the modulus constraints (Schaffalitzky 2000) can be used to eliminate several spurious solutions, reducing the number of possible solutions to 21.

When more than three views are present, the modulus constraints from all the views may be used in a least squares framework for greater accuracy and robustness:

$$\min_{p_1, p_2, p_3} \sum_{i=1}^{m} (\gamma_i \alpha_i^3 - \beta_i^3)^2. \tag{12}$$
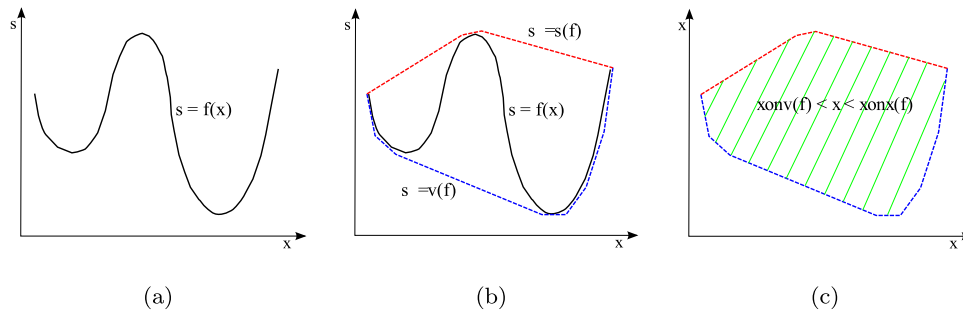
Fig. 2 (**a**) The objective function of the non-linear least squares problem $\min(f(x) - \mu)^2$ is linearized by replacing the non-linear function $f(x)$ by a scalar variable s and introducing an equality constraint $s = f(x)$. (**b**) Convex and concave relaxations are constructed for the function $f(x)$. For the functions encountered in this paper, Appendix A demonstrates the construction of tight, piecewise linear relaxations. (**c**) Instead of being limited to the curve $s = f(x)$, the constraint set is expanded into the relaxed region between the convex under-estimator and the concave over-estimator

A gradient-based optimization routine, such as (Levenberg 1944; Marquardt 1963), may be used to obtain a locally optimal solution to the above problem. In (Pollefeys and Gool 1999), several random initializations for the Levenberg-Marquardt algorithm are used to enhance the chances of converging to a global optimum.

6.2 Problem Formulation

Note that the cost function in (12) is a polynomial and some recent work in computer vision (Kahl and Henrion 2005; Chandraker et al. 2007a) exploits convex linear matrix inequality (LMI) relaxations to achieve global optimality in polynomial programs. However, this is a degree 8 polynomial in three variables, which is far beyond what present-day solvers can handle (Henrion and Lasserre 2003; Prajna et al. 2002).

We instead consider the equivalent formulation:

$$\min_{p_1, p_2, p_3} \sum_{i=1}^{m} (\gamma_i^{1/3} \alpha_i - \beta_i)^2, \tag{13}$$

for which the global minimum is estimated using the method outlined in this section. Please refer to Sect. 9 for further discussions on this reformulation.

6.3 Convex Relaxation

As an illustration of higher-level concepts, we show construction of convex under-estimators for the non-convex objective in (13). The actual objective we minimize incorporates chirality bounds and is derived in Sect. 6.4.

Let us suppose it is possible to derive a convex under-estimator $\text{conv}(\gamma_i^{1/3} \alpha_i)$ and concave over-estimator $\text{conc}(\gamma_i^{1/3} \alpha_i)$ for $\gamma_i^{1/3} \alpha_i$. Then the following convex op-

timization problem underestimates the solution to (13):

$$\min_{p_1, p_2, p_3} \sum_{i=1}^{m} (s_i - \beta_i)^2 \tag{14}$$

subject to $\text{conv}(\gamma_i^{1/3} \alpha_i) \leq s_i \leq \text{conc}(\gamma_i^{1/3} \alpha_i)$.

As shown in Appendix A.3, our convex and concave relaxations for functions of the form $x^{1/3}y$ are piecewise linear and representable using a small set of linear inequalities. Thus the above optimization problem is a convex quadratic program that can be solved using a quadratic programming (QP) or a second order cone programming (SOCP) solver.

Given bounds on $\{p_1, p_2, p_3\}$, a branch and bound algorithm can now be used to obtain a global minimum to the modulus constraints. All that remains to be shown is that it is possible to estimate an initial region which bounds the coordinates of π_∞.

6.4 Incorporating Bounds on the Plane at Infinity

One way to derive bounds on the coordinates of the plane at infinity is by using the chirality conditions overviewed in Sect. 2.3. Let $\mathbf{v}$ be the plane at infinity in the centered quasi-affine frame, where $\mathbf{v} = (v_1, v_2, v_3, 1)^\top$, so that we can find bounds on each v_i. However, the modulus constraints require that the first metric camera be of the form $\text{K}[\text{I}|\mathbf{0}]$ and the first projective camera have the form $[\text{I}|\mathbf{0}]$, which might not be satisfiable in a centered quasi-affine frame, in general. Thus, we need to use the bounds derived in the centered quasi-affine frame within the modulus constraints for the original projective frame.

The centered quasi-affine reconstruction differs from the projective one by a transformation $\text{H}_{qa} = \text{H}_a \text{H}_q$, where H_q takes the projective frame to some quasi-affine frame and H_a is the affine centering in that quasi-affine frame. Let $\mathbf{h}_i$ be the i-th column of H_{qa}, then we have $p_i = \mathbf{h}_i^\top \mathbf{v} / \mathbf{h}_4^\top \mathbf{v}$. Recall

that, for the j-th view, α_j, β_j and γ_j are affine expressions in p_1, p_2 and p_3 (Pollefeys and Gool 1999). Then, for instance,

$$\alpha_j = \alpha_{j1} p_1 + \alpha_{j2} p_2 + \alpha_{j3} p_3 + \alpha_{j4} \tag{15}$$

$$= \frac{a_j(\mathbf{v})}{d(\mathbf{v})}, \tag{16}$$

where, $a_j(\mathbf{v}) = \alpha_{j1}\mathbf{h}_1^\top \mathbf{v} + \alpha_{j2}\mathbf{h}_2^\top \mathbf{v} + \alpha_{j3}\mathbf{h}_3^\top \mathbf{v} + \alpha_{j4}\mathbf{h}_4^\top \mathbf{v}$ and $d(\mathbf{v}) = \mathbf{h}_4^\top \mathbf{v}$. Similarly, let

$$\beta_j = \frac{b_j(\mathbf{v})}{d(\mathbf{v})}, \tag{17}$$

$$\gamma_j = \frac{c_j(\mathbf{v})}{d(\mathbf{v})}, \tag{18}$$

where $a_j(\mathbf{v}), b_j(\mathbf{v}), c_j(\mathbf{v}), d(\mathbf{v})$ are linear functions of $\mathbf{v}$. In the following, for the sake of brevity, we will drop the reference to $\mathbf{v}$ and just use a_j, b_j, c_j, d. Now the optimization problem (13) can be rewritten as

$$\min_{v_1, v_2, v_3} \frac{\sum_{j=1}^{m} (c_j^{1/3} a_j - d^{1/3} b_j)^2}{d^{8/3}} \tag{19}$$

subject to $l_i \leq v_i \leq u_i, \quad i = 1, 2, 3.$

Introducing new scalar variables for some of the non-linear terms, the above is equivalent to

$$\min_{v_1, v_2, v_3} r$$

subject to $r \cdot e \geq \sum_{j=1}^{m} (f_j - g_j)^2,$

$$f_j = c_j^{1/3} a_j, \quad j = 1, \ldots, m, \tag{20}$$

$$g_j = d^{1/3} b_j, \quad j = 1, \ldots, m,$$

$$e = d^{8/3},$$

$$l_i \leq v_i \leq u_i, \quad i = 1, 2, 3.$$

As outlined in our general recipe for constructing convex relaxations (Sect. 5), we have reduced the non-convexity in the above optimization problem to a set of equality constraints. The quadratic inequality constraint is convex and is known as a rotated cone (Boyd and Vandenberghe 2004). Given bounds on v_i, it is easy to calculate bounds on a_j, b_j, c_j, d, by solving eight linear programs in three variables. Given these bounds, we can construct convex and concave envelopes of the non-linear functions e, f_j, g_j and use them to construct the following convex program that underestimates

the minimum of the problem (20):

$$\min_{v_1, v_2, v_3} r$$

subject to $r \cdot e \geq \sum_{j=1}^{m} (f_j - g_j)^2,$

$$\text{conv}(c_j^{1/3} a_j) \leq f_j \leq \text{conc}(c_j^{1/3} a_j),$$

$$j = 1, \ldots, m, \tag{21}$$

$$\text{conv}(d^{1/3} b_j) \leq g_j \leq \text{conc}(d^{1/3} b_j),$$

$$j = 1, \ldots, m,$$

$$e \leq \text{conc}(d^{8/3}),$$

$$l_i \leq v_i \leq u_i, \quad i = 1, 2, 3.$$

Notice that the convex envelope of $d^{8/3}$ is not needed. Since (21) is a minimization problem, e always takes its maximum possible value and does not require a lower bound.

Following Appendix A, our convex relaxation in (21) consists of a linear objective subject to linear and SOCP constraints, which can be efficiently minimized (Sturm 1999). A branch and bound algorithm can now be used to obtain an estimate of $\{v_1, v_2, v_3\}$, which globally minimizes the modulus constraints. Thereafter, the plane at infinity in the projective frame can be recovered as $\pi_\infty = \mathrm{H}_{qa}^\top \mathbf{v}$, which completes the projective to affine upgrade.

7 Globally Optimal Metric Upgrade

7.1 Traditional Solution

Recall that when the camera intrinsic parameters are held constant, the DIAC satisfies the infinite homography relations $\omega^* = \mathrm{H}_\infty^i \omega^* \mathrm{H}_\infty^{i\top}$, $i = 1, \ldots, m$, where equality holds up to a scale factor. The standard technique for estimating the DIAC is to first normalize the infinite homography matrix by dividing it by the cube-root of its determinant:

$$\widehat{\mathrm{H}} = \frac{\mathrm{H}}{\sqrt[3]{\mathrm{H}}}. \tag{22}$$

Since this normalization "equates" the scale on the two sides of the infinite homography relation, estimating the DIAC can now be posed as a least squares problem:

$$\min_{\omega^*} \sum_{i=1}^{m} \|\omega^* - \widehat{\mathrm{H}}_\infty^i \omega^* \widehat{\mathrm{H}}_\infty^{i\top}\| \tag{23}$$

which is typically solved linearly by ignoring the positive semidefiniteness requirement on the DIAC. For the cases where the linear solution does not yield a positive semidefinite DIAC, the closest positive semidefinite matrix is

estimated as a post-processing step by dropping the negative eigenvalues. It is well-documented that this may lead to a spurious calibration in practice (Hartley and Zisserman 2004).

7.2 Problem Formulation

For the optimal solution to the infinite homography relation, we note that both ω^* and H^i_∞ are homogeneous entities, so our cost function must correctly account for the scale factor before it can be used to search for the optimal DIAC. Moreover, the optimization algorithm itself must take into account the positive semidefiniteness of the DIAC.

A necessary condition for the matrix ω^* to be interpreted as $\omega^* = KK^\top$ is that $\omega^*_{33} = 1$. Thus, we fix the scale in the infinite homography relation by demanding that both the matrices on the left and the right hand side of the relation have their $(3, 3)$ entry equal to 1. To this end, we introduce additional variables λ_i and pose the minimization problem:

$$\min_{\omega^*, \lambda_i} \sum_i \left\| \omega^* - \lambda_i H^i_\infty \omega^* H^{i\top}_\infty \right\|^2_F$$

subject to $\omega^*_{33} = 1$,

$$\lambda_i \mathbf{h}^{i\top}_3 \omega^* \mathbf{h}^i_3 = 1, \qquad (24)$$

$$\omega^* \succ 0,$$

$$\omega^* \in \mathcal{D}.$$

Here, $\mathbf{h}^i_3$ denotes the third row of the 3×3 infinite homography H^i_∞ and $\mathcal{D}$ is some initial convex region whose choice is elucidated later in this section. For the present, it suffices to understand that the individual entries of ω^* lie within the convex region $\mathcal{D}$.

7.3 Convex Relaxation

We begin by introducing a new set of variables $\nu_i = \lambda_i \omega^*$. Here each matrix ν_i is a symmetric 3×3 matrix with entries $\nu_{ijk} = \lambda_i \omega^*_{jk}$. Also let us assume that the domain $\mathcal{D}$ is given in the form of bounds $[l_{jk}, u_{jk}]$ on the five unknown symmetric entries ω^*_{jk} of ω^*. Then (24) can be re-written as

$$\min_{\omega^*, \nu_i, \lambda_i} \sum_i \left\| \omega^* - H^i_\infty \nu_i H^{i\top}_\infty \right\|^2_F$$

subject to $\nu_{ijk} = \lambda_i \omega^*_{jk}$,

$$\omega^*_{33} = 1, \qquad (25)$$

$$\lambda_i \mathbf{h}^{i\top}_3 \omega^* \mathbf{h}^i_3 = 1,$$

$$\omega^* \succ 0,$$

$$l_{jk} \le \omega^*_{jk} \le u_{jk}.$$

The non-convexity in the above optimization problem has been reduced to the bilinear equality constraints $\nu_{ijk} = \lambda_i \omega^*_{jk}$ and $\lambda_i \mathbf{h}^{i\top}_3 \omega^* \mathbf{h}^i_3 = 1$.

Given bounds on the entries of ω^*, a relaxation of (25) is obtained by replacing the constraint $\lambda_i \mathbf{h}^{i\top}_3 \omega^* \mathbf{h}^i_3 = 1$ by a pair of linear inequalities of the form $L_i \le \lambda_i \le U_i$, where L_i and U_i are computed by simply inverting the bounds on $\mathbf{h}^{i\top}_3 \omega^* \mathbf{h}^i_3$. Thus, the lower bound L_i can be computed as the reciprocal of the result of the maximization problem:

$$\max_{\omega^*} \mathbf{h}^{i\top}_3 \omega^* \mathbf{h}^i_3$$

subject to $\omega^*_{33} = 1$,

$$\omega^* \succeq 0, \qquad (26)$$

$$l_{jk} \le \omega^*_{jk} \le u_{jk}.$$

This is a semi-definite program (SDP) in 9 variables and can be solved very efficiently using interior point methods (Boyd and Vandenberghe 2004). The upper bound U_i can be computed similarly by computing the reciprocal of the minimizer of the above. The relaxed optimization problem can now be stated as:

$$\min_{\omega^*, \nu_i, \lambda_i} \sum_i \left\| \omega^* - H^i_\infty \nu_i H^{i\top}_\infty \right\|^2_F$$

subject to $\nu_{ijk} = \lambda_i \omega^*_{jk}$,

$$\omega^*_{33} = 1,$$

$$\omega^* \succ 0, \qquad (27)$$

$$l_{jk} \le \omega^*_{jk} \le u_{jk},$$

$$L_i \le \lambda_i \le U_i.$$

In effect, the above ensures that the introduction of an additional view does not translate into an increase in the dimensionality of our search space. Instead, the cost is limited to solving a small SDP to compute bounds on λ_i, while the branching variables remain the five unknowns of ω^*. Thus, the search space for the branch and bound algorithm can be restricted to a small, fixed number of dimensions, independent of the number of views.

Appendix A.2 discusses the synthesis of convex relaxations of bilinear equalities, which allows us to replace each bilinear equality by a set of linear inequalities. Using them, a convex relaxation of the above optimization problem can

be stated as

$$\min_{\omega^*, v_i, \lambda_i} \sum_i \left\| \omega^* - H_\infty^i v_i H_\infty^{i\top} \right\|_F^2$$

subject to $v_{ijk} \leq U_i \omega_{jk}^* + l_{jk} \lambda_i - U_i l_{jk},$

$$v_{ijk} \leq L_i \omega_{jk}^* + u_{jk} \lambda_i - L_i u_{jk},$$

$$v_{ijk} \geq L_i \omega_{jk}^* + l_{jk} \lambda_i - L_i l_{jk},$$

$$v_{ijk} \geq U_i \omega_{jk}^* + u_{jk} \lambda_i - U_i u_{jk}, \quad (28)$$

$$\omega_{33}^* = 1,$$

$$\omega^* \succeq 0,$$

$$l_{jk} \leq \omega_{jk}^* \leq u_{jk},$$

$$L_i \leq \lambda_i \leq U_i.$$

The objective function of the above optimization problem is convex quadratic. The constraint set includes linear inequalities and a positive semi-definiteness constraint. Such problems can be efficiently solved to their global optimum using interior point methods and a number of software packages exist for doing so. We use SeDuMi in our implementation (Sturm 1999).

The user of the algorithm specifies valid ranges for the entries of the calibration matrix K. From this input, we derive intervals $[l_{jk}, u_{jk}]$ for the entries ω_{jk}^* of the matrix ω^* using the rules of interval arithmetic (Moore 1966), which specifies the initial convex region $\mathcal{D}$ in (24).

7.3.1 A Note on Normalization

The careful reader would have observed that we do not follow the standard prescription of normalizing the infinite homography by the cube root of its determinant, as discussed in Sect. 7.1, to resolve the scale factors in the infinite homography relations $\omega^* = \lambda_i H_\infty^i \omega^* H_\infty^{i\top}, \ i = 1, \ldots, m.$

Since the equation we are trying to satisfy with the optimal estimate of ω^* is algebraic, there are a number of ways in which the scale ambiguity can be resolved. The method based on normalizing the determinant is just one of them. Note that the DIAC is not an arbitrary 3×3 symmetric positive definite matrix, it has a particular geometric and numerical interpretation, which requires that $\omega_{33}^* = 1$. Our choice of the objective function reflects this. Even in the traditional approach, while normalizing the determinant technically resolves the scale ambiguity of the infinite homography relation, a further normalization would be needed before the intrinsic calibration matrix K can be estimated from the DIAC.

Since we explicitly optimize over the λ_i variables also, our approach subsumes the standard one which just corresponds to a priori setting $\lambda_i = \det(H_\infty^i)^{-2/3}$ in the objective function of (24) and performing the minimization only

with respect to ω^*. In general, the solution returned by our method that optimizes over both ω^* and the λ_i's, will correspond to a minimum which is lower than the one that corresponds to the standard normalization. In other words, our approach is more consonant with the aim of global optimization because we *estimate* all the scale factors, while the traditional approach *chooses* them. Note that if the choices of λ_i corresponding to the traditional normalization fortuitously happen to be the optimal ones, our algorithm will find them.

Thus, our method, at the expense of some computational effort, poses the optimization problem in terms of an interpretable quantity and finds estimates which are at least as good as, or better than, those obtained by using the standard normalization.

8 Experiments

In this section, we will describe the experimental evaluation of our algorithms using synthetic and real data.

To evaluate the output of our algorithm the following metrics are defined:

$$\Delta p = \sqrt{\sum_{i=1}^3 \left(p_i / p_i^0 - 1 \right)^2} \quad (29)$$

$$\Delta f = \left| \frac{f_1}{f_1^0} - 1 \right| + \left| \frac{f_2}{f_2^0} - 1 \right| \quad (30)$$

$$\Delta uv = |u - u^0| + |v - v^0| \quad (31)$$

$$\Delta s = |s - s^0| \quad (32)$$

Here, p_i are estimated coordinates of the plane at infinity, f_1, f_2 represents the two focal lengths, (u, v) stands for the principal point and s for the skew. $p_i^0, f_1^0, f_2^0, u^0, v^0$ and s^0 are the corresponding ground truth quantities.

In the first experiment, we simulated a scene where 100 3D points are randomly generated in a cube with sides of length 20, centered at the origin and a varying number of cameras are randomly placed at a nominal distance of 40 units. Zero mean, Gaussian noise of varying standard deviation is added to the image coordinates. A projective transformation is applied to the scene with a known, randomly generated plane at infinity and the ground truth intrinsic calibration matrix is identity. All the statistics reported in this section are acquired over 50 trials.

Table 1 reports, for various number of cameras and noise levels, the errors in the estimates of various camera parameters and the number of iterations needed for the algorithm to converge. The termination criterion measures the gap, ϵ, between the lowest lower bound and the current best objective function value. The column $\pi_\infty(1)$ in Table 1 reports the

Table 1 Error in camera calibration parameters for random synthetic data. The errors in the intrinsic parameters are reported relative to ground truth for the indicated quantities. Mean and standard deviations are reported for the number of iterations. All quantities reported are averaged over 50 trials. σ stands for percentage noise in image coordinates and m stands for number of views

σ (%)	m	Error				Iterations		
		Δp	Δf	Δuv	Δs	$\pi_\infty(1)$	$\pi_\infty(2)$	ω^*
0	5	3.65e–5	3.28e–5	3.01e–5	2.89e–5	31.1 ± 16.2	11.02 ± 7.96	32.1 ± 11.6
	10	2.51e–6	2.14e–6	2.03e–6	2.00e–6	18.7 ± 11.8	4.19 ± 2.26	40.9 ± 12.5
	20	1.28e–6	1.51e–6	1.33e–6	1.73e–6	21.4 ± 12.5	1.62 ± 0.81	31.8 ± 10.4
	40	9.08e–7	8.24e–7	7.99e–7	7.58e–7	23.8 ± 10.8	1.14 ± 0.45	27.5 ± 14.3
0.1	5	4.76e–4	4.59e–4	4.22e–4	4.05e–4	27.3 ± 15.1	9.93 ± 6.41	36.3 ± 12.1
	10	3.44e–4	3.07e–4	2.73e–4	2.91e–4	17.2 ± 9.0	3.49 ± 2.33	44.2 ± 15.7
	20	2.75e–4	2.92e–4	2.56e–4	2.31e–4	16.1 ± 8.7	2.35 ± 1.94	33.0 ± 12.3
	40	2.55e–4	2.41e–4	2.06e–4	1.85e–4	23.2 ± 11.6	7.95 ± 5.79	30.1 ± 10.6
0.2	5	1.19e–3	1.14e–3	9.92e–4	8.73e–4	41.0 ± 24.5	12.46 ± 8.39	38.4 ± 13.7
	10	7.65e–4	7.13e–4	7.01e–4	6.85e–4	24.7 ± 14.9	4.45 ± 2.56	47.9 ± 16.9
	20	6.03e–4	6.80e–4	5.12e–4	5.79e–4	19.5 ± 14.4	6.99 ± 5.41	34.5 ± 12.8
	40	5.59e–4	6.05e–4	4.29e–4	5.10e–4	33.2 ± 18.3	10.60 ± 6.47	31.6 ± 12.1
0.5	5	3.29e–3	3.22e–3	3.02e–3	2.63e–3	63.2 ± 26.7	11.58 ± 8.18	42.7 ± 18.5
	10	1.66e–3	2.19e–3	1.99e–3	2.11e–3	29.8 ± 12.7	6.23 ± 4.75	51.1 ± 17.2
	20	1.41e–3	1.84e–3	1.43e–3	1.55e–3	22.3 ± 11.9	8.23 ± 5.35	38.2 ± 12.9
	40	1.25e–3	1.50e–3	1.18e–3	9.06e–4	46.6 ± 19.0	20.60 ± 7.23	32.4 ± 13.4
1.0	5	4.68e–3	4.04e–3	3.77e–3	3.36e–3	74.1 ± 28.1	9.86 ± 8.42	45.5 ± 17.2
	10	3.15e–3	2.88e–3	2.52e–3	2.15e–3	36.4 ± 14.2	9.23 ± 7.17	56.8 ± 20.6
	20	2.86e–3	2.45e–3	2.02e–3	1.74e–3	31.0 ± 12.8	15.75 ± 7.30	40.9 ± 14.0
	40	2.79e–3	2.21e–3	1.76e–3	1.30e–3	56.4 ± 19.8	23.56 ± 8.34	38.9 ± 14.3

number of branch and bound iterations using the algorithm described in 6.4. However, an additional optimization is possible: we can refine the value of the feasible point $f(q^*)$ using a gradient descent method within the rectangle that contains it. This does not compromise optimality, but allows the value of the current best estimate to be lower than the value corresponding to the minimum of the lower bounding function. The number of iterations with this refinement is tabulated under $\pi_\infty(2)$. The error metrics reported are computed using the refined algorithm, however, since both algorithms are run with the same very stringent tolerance ($\epsilon = 1e$–7), the solutions obtained are comparable. The number of iterations for the DIAC estimation (with no refinement) are tabulated under ω^*. The metric upgrade step was performed with $\epsilon = 1e$–5. The termination criterion measures the gap, ϵ, between the lowest lower bound and the current best objective function value.

Figure 3 plots the errors graphically. The accuracy of the algorithm is evident from the very low error rates obtained for reasonable noise levels. It is interesting that the algorithm performs quite well even for noise as high as 1%. In general, the accuracy improves as expected when more cameras are used.

To demonstrate scalable runtime behavior, Fig. 4 plots the runtime for the affine and metric upgrade stages for the random data experiment with 0.1% noise, for varying numbers of cameras. These experiments were conducted on a Pentium IV, 1 GHz computer with 1 GB of RAM. Note that the graceful variation in the runtime behavior is a direct outcome of our bounds propagation schemes, without which the branch and bound algorithms would display exponential characteristics. Our code is un-optimized MATLAB with an off-the-shelf SDP solver (Sturm 1999), so the actual magnitude of these timings should be understood only as rough qualitative indicators.[1]

While the metrics in Table 1 are intuitive for evaluating the intrinsic parameters, it is not readily evident how Δp should be interpreted. Towards that end, we perform a set of experiments, inspired by (Pollefeys and Gool 1999), where three mutually orthogonal 5×5 grids are observed by varying numbers of randomly placed cameras. Noise ranging from 0.1 to 1% is added to the image coordinates. The quality of the affine upgrade is indirectly inferred from the

---

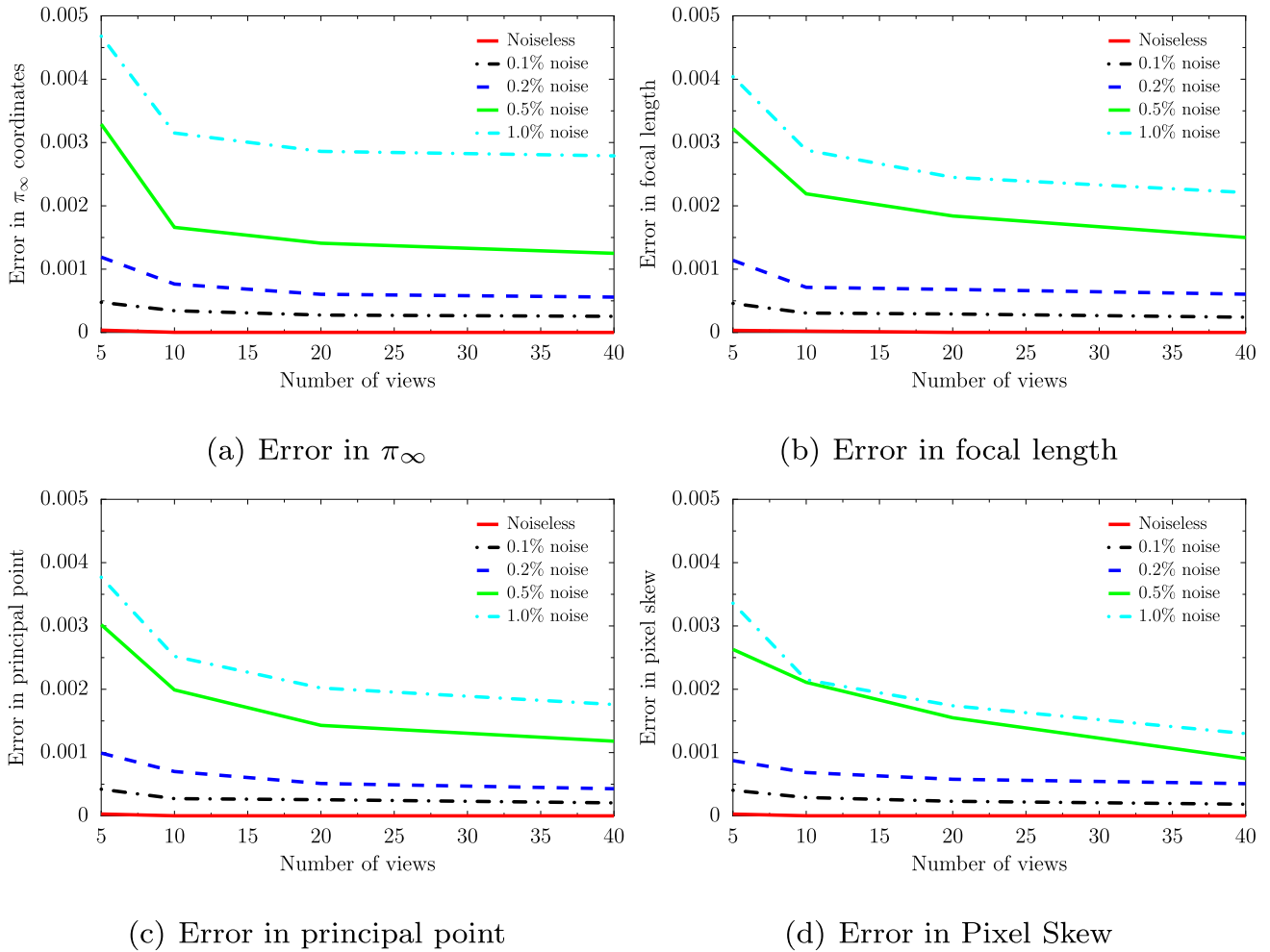[1]Prototype code available at http://vision.ucsd.edu/stratum.

Fig. 3 Error in camera calibration parameters for random synthetic data. The errors in the graphs are plotted relative to ground truth for the indicated quantities. All quantities reported are averaged over a 50 trials
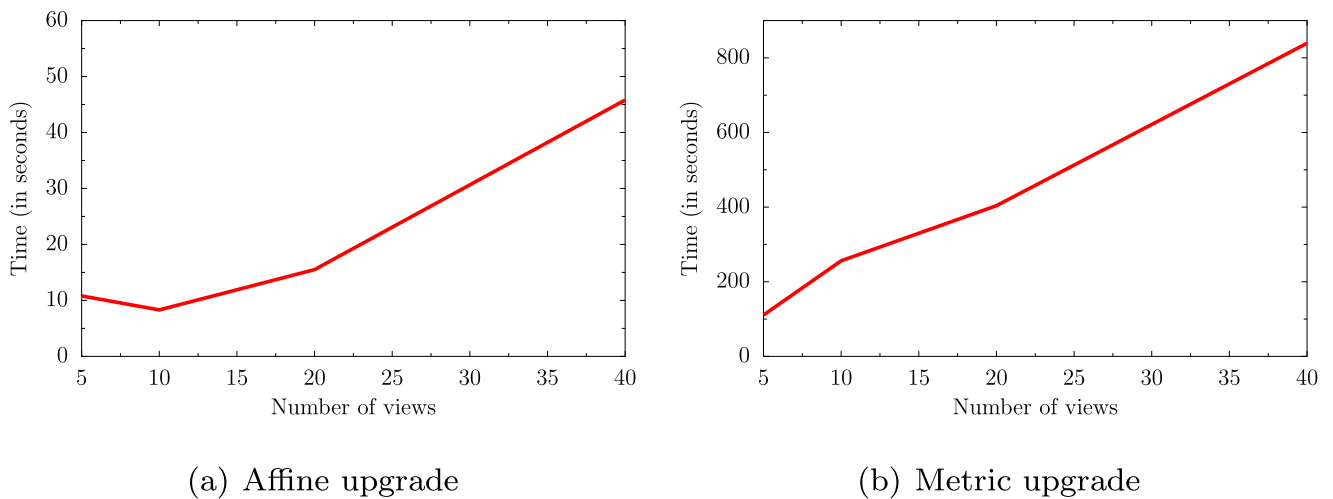
(a) Error in π_∞

(b) Error in focal length

(c) Error in principal point

(d) Error in Pixel Skew

Fig. 4 Runtime behavior of the branch and bound algorithms for the affine and metric upgrade steps. All timings reported are averaged over 50 trials

(a) Affine upgrade

(b) Metric upgrade

Table 2 Error in affine and metric properties for three synthetically generated, mutually orthogonal planar grids. The table reports mean angular deviations from parallelism and orthogonality, measured in degrees. All quantities reported are averaged over 50 trials

Noise (%)	Views	Affine (Parallel)	Metric (Parallel)	(Perpendicular)
0	5	2.40e–6	2.46e–6	3.07e–6
	10	7.90e–7	8.12e–7	1.12e–6
	20	5.22e–7	5.50e–7	8.70e–7
	40	3.67e–7	3.88e–7	6.23e–7
0.1	5	0.40	0.40	0.34
	10	0.27	0.27	0.23
	20	0.19	0.19	0.15
	40	0.13	0.13	0.10
0.2	5	0.79	0.80	0.63
	10	0.54	0.54	0.44
	20	0.36	0.36	0.25
	40	0.25	0.25	0.19
0.5	5	1.95	1.96	1.88
	10	1.31	1.31	1.02
	20	0.89	0.89	0.79
	40	0.64	0.64	0.57
1.0	5	4.05	4.07	3.97
	10	2.63	2.63	2.30
	20	1.83	1.83	1.52
	40	1.27	1.27	1.09
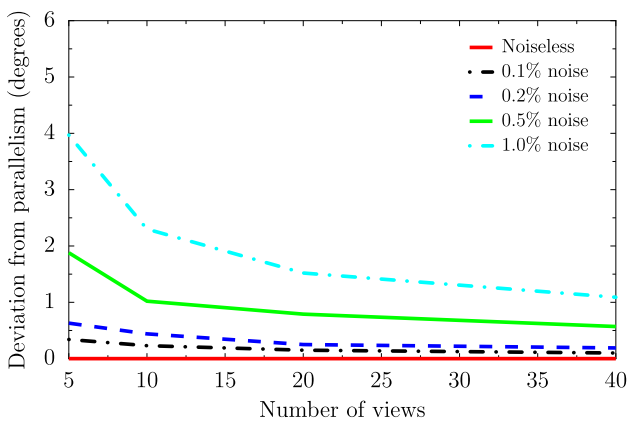
(a) Deviation from parallelism
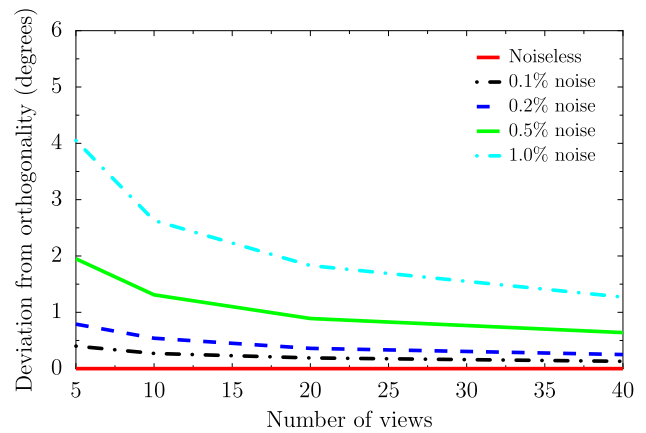
(b) Deviation from orthogonality

Fig. 5 Errors in affine and metric properties for three synthetically generated, mutually orthogonal planar grids. The graphs plot (**a**) angular deviation from parallelism after the affine upgrade and (**b**) angular deviation from orthogonality after the metric upgrade, both measured in degrees. All quantities reported are averaged over 50 trials

deviation from parallelism in the reconstructed grid lines, while the quality of the metric upgrade is inferred from the deviation from orthogonality. Table 2 reports the results of this experiment and Fig. 5 shows the results graphically.

Again, we observe that the algorithm achieves very good accuracy for reasonable noise and performs quite well even for 1% noise. With just 5 cameras, it is quite likely for the configuration to be ill-conditioned or degenerate, which causes the algorithm to break down in some cases.

We also use this experimental setup to compare against traditional local optimization approaches. In the first set of experiments, for a fixed number of views ($m = 20$) and vary-
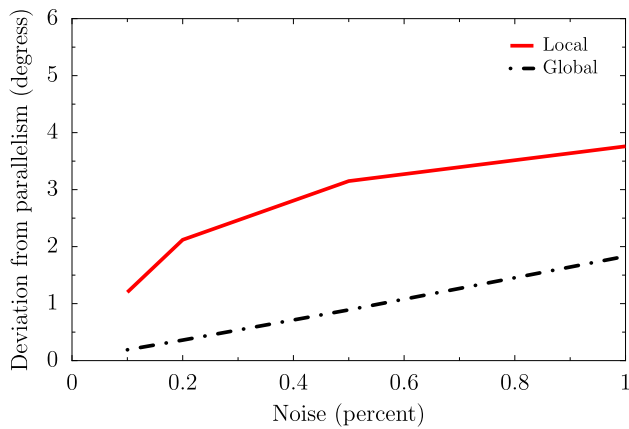
Fig. 6 Comparison of the accuracy of the plane at infinity estimated by the globally optimal method (*black dotted curve*) and a Levenberg-Marquardt routine with 50 random initializations (*red curve*). The accuracy of the affine upgrade is deduced from the extent to which parallelism is preserved in the reconstruction

Fig. 7 Comparison of the globally optimal metric upgrade algorithm with a traditional linear method for estimating the DIAC. The accuracy of the reconstruction is deduced by the extent to which orthogonality is preserved in the metric reconstruction. The *solid red curve* plots the deviation from orthogonality when a local optimization method is used for estimating the plane at infinity and a linear method is used for estimating the DIAC. The *dotted black curve* uses a linear method for DIAC estimation, but the optimal plane at infinity is used for affine upgrade. The *dashed blue curve* uses optimal algorithms for both the affine and metric upgrade steps. The reported quantities are averages of 50 trials

ing noise levels, we minimize the modulus constraints in (13) using a Levenberg-Marquardt scheme with 50 random initializations. It can be seen from Fig. 6 that the globally optimal solution of Sect. 6 yields more accurate solutions.

Finally, once the plane at infinity has been estimated, the globally optimal metric upgrade of Sect. 7 is compared against the traditional linear approach. The drawbacks of the traditional linear approach discussed previously, namely, overlooking the positive semidefiniteness requirement of the DIAC and choosing a suboptimal scale factor, are illustrated by these experiments.

For a fixed number of views ($m = 5$), the deviation from orthogonality observed in the globally optimal metric reconstruction is compared to the deviation obtained from the linear method, across varying noise levels (Fig. 7). Two sets of metric reconstructions are performed for the linear method. For the first set, the metric upgrade is performed starting with the affine reconstruction obtained using the plane at infinity estimated by a local optimization method (Levenberg-Marquardt with 50 random initializations). For the second set, the affine upgrade is computed using the optimal method in Sect. 6, while the metric upgrade is linear. For any cases where the estimated DIAC is not positive semidefinite, the calibration matrix must be extracted after a further approximation to project on the closest point on the cone of positive semidefinite matrices. Clearly, the globally optimal DIAC estimation outperforms the linear method across all noise levels, which shows the importance of estimating the correct normalization factors λ_i introduced in Sect. 7.2.

An important consideration for noisy situations is the sensitivity of the chirality bounds to outliers. Similar to (Nistér 2004), for noisy images expected in a real world scenario, chirality bounds are computed using only the camera centers. The reason is that usually there are far more points
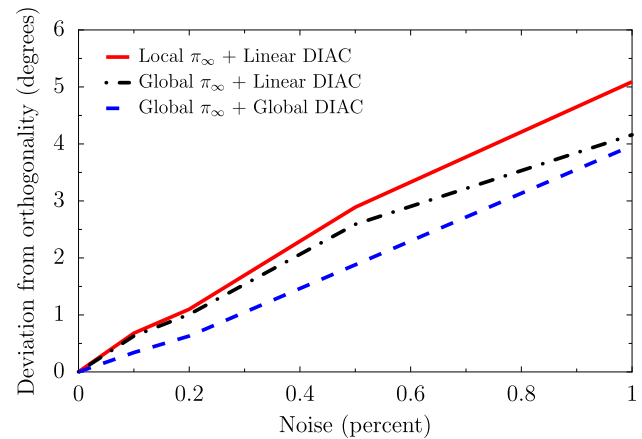
than cameras and the camera centers are likely to be estimated more robustly than 3D points.

To demonstrate performance on real data, we consider images of marker targets on two orthogonal walls (Fig. 8(a)). Using image correspondences from detected corners, we perform a projective reconstruction using the projective factorization algorithm (Sturm and Triggs 1996) followed by bundle adjustment. The normalization procedure and exact implementation follows the description in (Hartley and Zisserman 2004).

Bounds on the plane at infinity are computed using chirality, c.f. (8). Focal lengths are assumed to lie in the interval [500, 1500], principal point within [250, 450] × [185, 385] (image size is 697 × 573) and skew [−0.1, 0.1]. The plane at infinity and DIAC are estimated using our algorithm. While ground truth measurements for the scene are not available, we can indirectly infer some observable properties.

The coplanarity of the individual targets is indicated by the ratio of the first eigenvalue of the covariance matrix of their points to the sum of eigenvalues. This ratio is measured to be 3.1×10^{-6}, 4.1×10^{-5}, 6.2×10^{-5} and 4.1×10^{-4} for the four polygonal targets. The angle between the normals to the planes represented by two targets on the adjacent walls is 88.1° in our metric reconstruction (Fig. 8(b)). The same angle is measured as 89.8° in a reconstruction using (Pollefeys et al. 2002). The precise ground truth angle between the targets is unknown.

To demonstrate the performance of the algorithm on another real dataset, we consider a corridor sequence, which

Fig. 8 In the reconstruction, targets on the same plane are represented by *lines* of the same color. The *second view* of the obtained 3D reconstruction shows that the angle between targets on adjacent walls is recovered as nearly 90 degrees

(a) Four of the twelve input images used for reconstruction
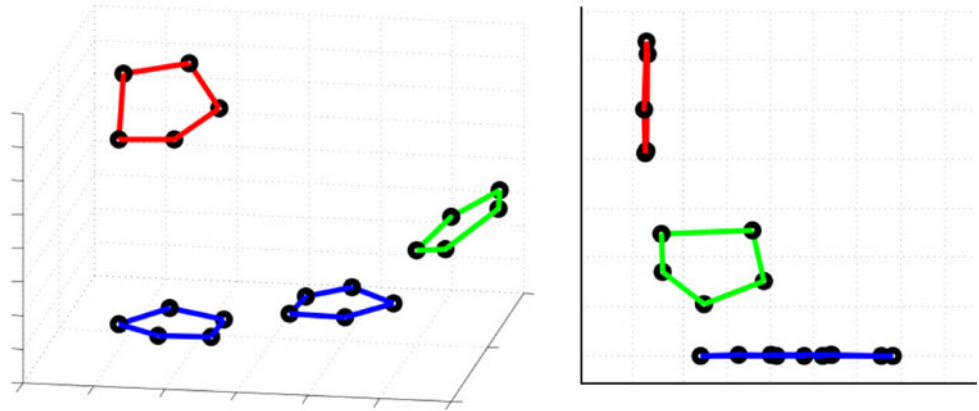
(b) Two views of the obtained 3D reconstruction

Fig. 9 Performance on the corridor sequence, with a nearly straight line camera motion. The camera trajectories obtained in metric reconstructions using various algorithms are plotted here. (**a**) "Ground truth" from bundle adjustment. (**b**) Output of the optimal stratified algorithm of this paper. (**c**) Output of linear algorithm for estimating the dual quadric

(a)   (b)   (c)
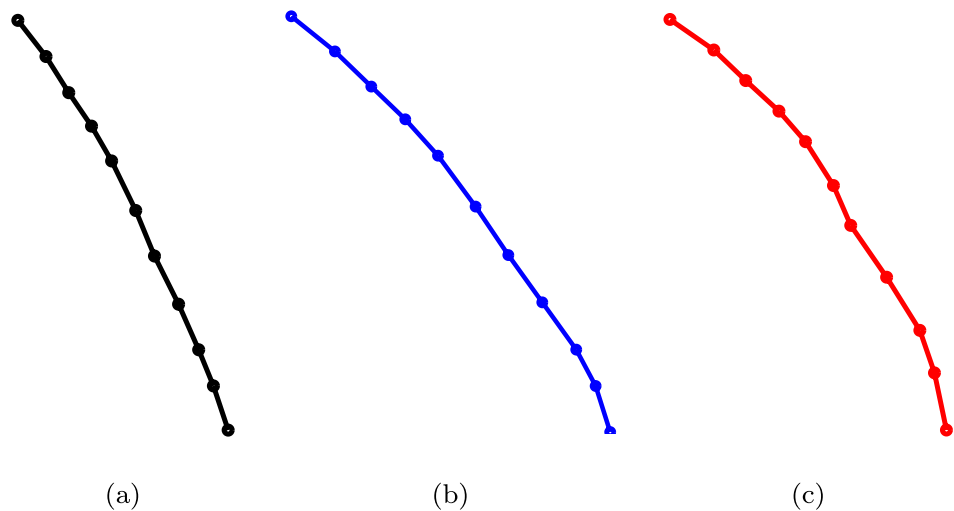
consists of 11 images acquired under a nearly straight line motion (Hartley and Zisserman 2004). Starting from a projective reconstruction, chirality bounds on the plane at infinity are computed using the camera centers and a few stably tracked points. The camera trajectory in the metric reconstruction using the stratified algorithm of this paper is shown in Fig. 9(b). The metric reconstruction using a locally optimal linear algorithm for directly estimating the absolute quadric (Pollefeys et al. 2002) is shown in Fig. 9(c). Although the camera motion in this sequence is close to a theoretically critical configuration, the global methods of this paper perform well, but the estimate from the linear method

is also quite reasonable. In particular, it can be seen that the output of the optimal stratified algorithm is quite close to the "ground truth" obtained by full bundle adjustment.

All the results that we have reported are for the raw output of our algorithm. In practice, a few iterations of bundle adjustment following our algorithms might be used to achieve a slightly better estimate.

9 Conclusions and Further Discussions

In this paper, we have presented globally optimal solutions to the affine and metric stages of stratified autocalibration.

Although our cost function is algebraic, this is the first work that provides optimality guarantees for scalable stratified autocalibration.

For the success of a branch and bound scheme, it is of utmost importance that the convex relaxations be as tight as possible. The second order cone programming based convex relaxation that we develop for solving the affine upgrade step and the semidefinite programming based convex relaxation for the metric upgrade step satisfy this requirement, while also being very fast to compute in practice.

Sometimes, a consideration of practicality of the convex relaxation influences the choice of the algebraic form of an objective function. Indeed, the most straightforward way to minimize the modulus constraints would be to use the simpler formulation of (12) and construct a multi-level relaxation for the quartic polynomials by successively using the bilinear relaxation of Sect. A.2. However, in our experience, such multistep relaxations are very loose in practice, so the branch and bound algorithm will not converge in a reasonable amount of time. Thus, the least squares version of the modulus constraints we globally minimize corresponds to the reformulated version of (13).

A crucial aspect of designing a global optimization algorithm based on branch and bound is the choice of initial region, which must be principled and guaranteed to contain the optimal solution. Arbitrarily choosing a very large initial region will lead to impractically long convergence times for the branch and bound, while too restrictive a choice might not contain the globally optimum point. Our affine upgrade step addresses this issue by incorporating chirality constraints within the convex relaxation for the modulus constraints. In practice, this limits the location of the plane at infinity to a small region of the search space. For the metric upgrade step, the entries of the DIAC that we wish to estimate are related to more tangible entities corresponding to the internal parameters of the camera. So, a user can easily specify reasonable bounds on the focal length, pixel skew and principal point, which are propagated to initial bounds on the DIAC using interval arithmetic.

Several important extensions to the methods introduced in this paper can be envisaged. For instance, an L_1-norm formulation will allow us to use an LP solver for the affine upgrade, making it possible to solve larger problems faster. To the best of our knowledge, it remains an open question whether methods similar to those proposed in this paper can be used for obtaining optimal solutions to the direct autocalibration problem. A global optimization approach for estimating the absolute dual quadric is discussed in (Chandraker et al. 2007a), however, it employs LMI relaxations and is not guaranteed to return a certificate of optimality.

Finally, we reiterate that pragmatic application of domain knowledge is important for successfully employing an optimization paradigm to globally optimize a computer vision problem. Indeed, it is the careful consideration of multiview geometry for choosing the initial region, constructing convex relaxations and restricting the dimensionality of the search space within a branch and bound framework that allows the globally optimal algorithms presented in this paper to be practical. We are hopeful that, in the near future, methods not unlike ours will be used to exploit underlying convexities to successfully optimize challenging problems in other areas of computer vision, besides multiview geometry.

Appendix A: Convex and Concave Relaxations

In this appendix, we briefly describe the convex and concave relaxations of the intermediate non-linear terms that were relaxed as part of the various convex relaxations in the main text. In each instance, the variables x and y take values in the intervals $[x_l, x_u]$ and $[y_l, y_u]$, respectively.

A.1 Functions of the Form $f(x) = x^{8/3}$

The function $x^{8/3}$ is convex, and thus the line joining $(x_l, x_l^{8/3})$ and $(x_u, x_u^{8/3})$ is a tight concave over-estimator, thus the relaxation is given by

$$z \le x_l^{8/3} + \frac{x - x_l}{x_u - x_l}(x_u^{8/3} - x_l^{8/3}). \tag{A.1}$$

A.2 Bilinear Functions $f(x, y) = xy$

We begin by considering convex and concave relaxations of the bilinear function $f(x) = xy$. It can be shown (Al-Khayyal and Falk 1983) that the tightest convex lower bound for $f(x, y)$ is given by the function
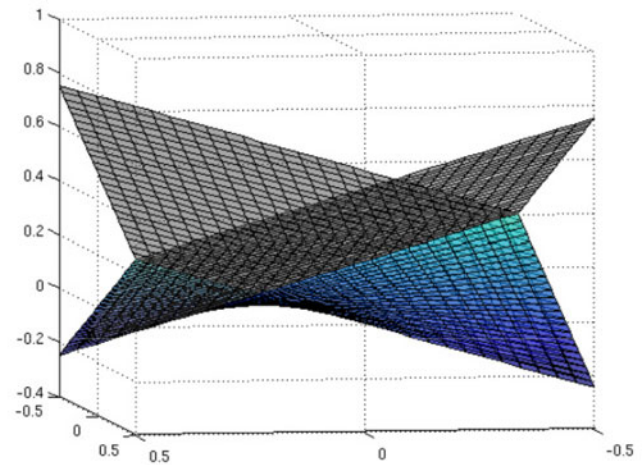
$$z = \max\{x_l y + y_l x - x_l y_l, x_u y + y_u x - x_u y_u\}. \tag{A.2}$$

Similarly, the tightest concave upper bounding function is given by

$$z = \min(x_u y + y_l x - x_u y_l, x_l y + y_u x - x_l y_u). \tag{A.3}$$
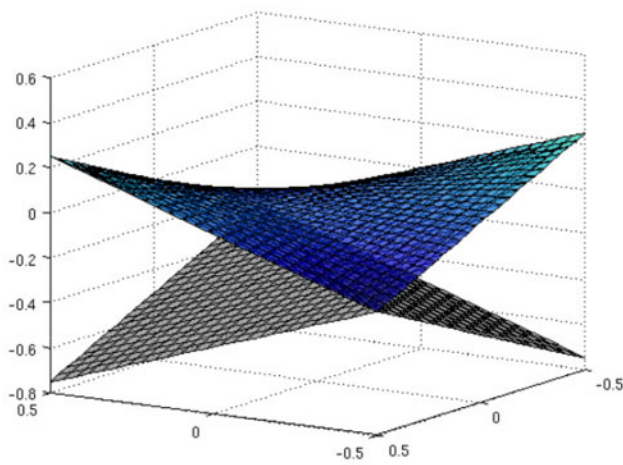
(a)   (b)

Fig. 10 (**a**) Construction of the convex relaxation for the bilinear function $f(x, y) = xy$. The function is drawn in the blue shade, while the convex envelope is the point wise maximum of the two *gray shaded planes*. (**b**) Construction of the concave relaxation for the bilinear func-

tion $f(x, y) = xy$. The function is drawn in the blue shade, while the concave envelope is the point wise minimum of the two *gray shaded planes*. A different view of the bilinear function is shown to better illustrate the relaxation

Thus, the convex relaxation of the equality constraint $z = xy$ over the domain $[x_l, y_l] \times [y_l, y_u]$ is given by

$$z \geq x_l y + y_l x - x_l y_l,$$
$$z \geq x_u y + y_u x - x_u y_u,$$
$$z \leq x_u y + y_l x - x_u y_l, \qquad (A.4)$$
$$z \leq x_l y + y_u x - x_l y_u.$$

These relaxations are graphically illustrated in Fig. 10.

A.3 Functions of the Form $f(x, y) = x^{1/3} y$

We now consider the construction of the convex relaxation for a bivariate function $f(x, y) = x^{1/3} y$. As illustrated in Fig. 11, this is a non-convex function whose convex relaxation is not straightforward.

Case I: $[x_l > 0 \ or \ x_u < 0]$

Suppose $x_l > 0$, then $f(x, y)$ is concave in x and convex in y. It can be shown (Tawarmalani and Sahinidis 2002) that the convex envelope for $f(x, y)$ is given by

$$\min z$$

subject to $z \geq (1 - \lambda) f(x_l, y_a) + \lambda f(x_u, y_b),$
$$x = x_l + (x_u - x_l)\lambda,$$
$$y = (1 - \lambda) y_a + \lambda y_b, \qquad (A.5)$$
$$0 \leq \lambda \leq 1,$$
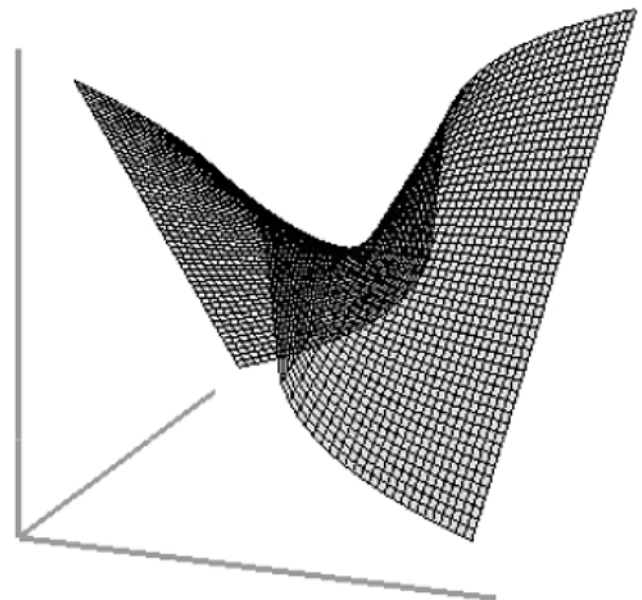$$y_l \leq y_a, y_b \leq y_u.$$

Fig. 11 The non-convex function $f(x, y) = x^{1/3} y$

Noting that $f(x, y) = x^{1/3} y$, substituting $y_p = (1 - \lambda) y_a$ and simplifying results in the following convex relaxation:

$$z \geq x_l^{1/3} y_p + x_u^{1/3}(y - y_p),$$
$$(1 - \lambda) y_l \leq y_p \leq (1 - \lambda) y_u,$$
$$\lambda y_l \leq y - y_p \leq \lambda y_u, \qquad (A.6)$$
$$\lambda = \frac{x - x_l}{x_u - x_l}.$$

Fig. 12 Construction of the concave overestimator for $x^{1/3}$, illustrated with the *solid, black curve*. (**a**) The concave overestimator when $-x_l/8 < x_u$ is given by the minimum of the *dashed red* and *blue lines*. (**b**) The concave overestimator when $-x_l/8 \geq x_u$ is given by the *dashed blue line*
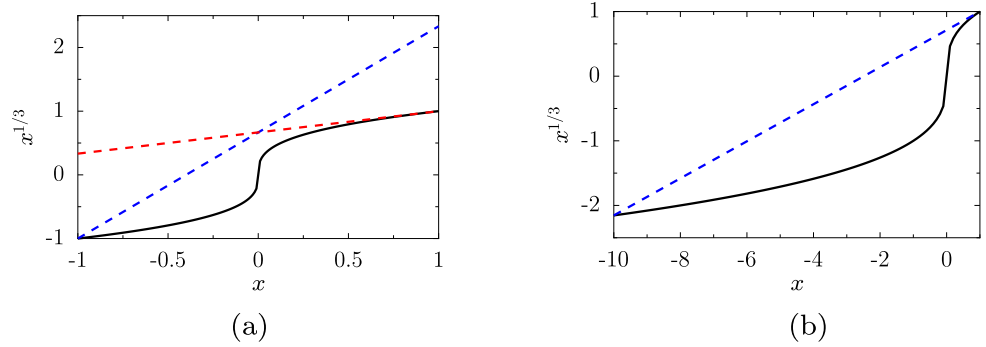
A concave relaxation for $x^{1/3}y$ can be constructed by considering the negative of the convex envelope of $x^{1/3}(-y)$. This leads to

$$z \leq (x_u^{1/3} - x_l^{1/3})y_p' + x_u^{1/3}y,$$

$$\lambda y_u \geq y + y_p' \geq \lambda y_l,$$

$$(1-\lambda)y_u \geq -y_p' \geq (1-\lambda)y_l, \tag{A.7}$$

$$\lambda = \frac{x - x_l}{x_u - x_l}.$$

For the case when $x_u < 0$, we observe that a convex relaxation for $x^{1/3}y$ is given by the negative of the concave relaxation of $t^{1/3}y$, where $t = -x$. Appropriate manipulation of (A.6) gives us the convex and concave envelopes for this case too.

Case II: [$x_l \leq 0 \leq x_u$]

The function $x^{1/3}$ is convex for $x < 0$ and concave for $x > 0$. The derivation in (A.6) depends critically on the convexity of $x^{1/3}$ over its domain, and thus, cannot be used for the case when $x_l \leq 0 \leq x_u$. So instead of a one step relaxation, we will consider the two equality constraints $t = x^{1/3}$, $z = ty$ and relax each of them individually. Once we have the relaxation for $t = x^{1/3}$, we can then apply the bilinear relaxation to $z = ty$.

To construct a concave overestimator for $x^{1/3}$, we notice that when $-x_l/8 < x_u$, a line which upper bounds the curve $x^{1/3}$ is the tangent at $(x_u, x_u^{1/3})$, given by

$$t = \frac{1}{3}x_u^{-2/3}x + \frac{2}{3}x_u^{1/3}. \tag{A.8}$$

While this line is a concave overestimator suitable for branch and bound in principle, we can make it a tighter relaxation. Notice that $x^{1/3}$ is concave in the region $(x_l, 0)$ and the line in (A.8) passes through $(0, \frac{2}{3}x_u^{2/3})$ which lies above the curve. Thus, the line segment joining $(x_l, x_l^{1/3})$ and $(0, \frac{2}{3}x_u^{2/3})$, given by

$$t = \left(x_l^{-2/3} - \frac{2}{3}x_l^{-1}x_u^{1/3}\right)x + \frac{2}{3}x_u^{1/3} \tag{A.9}$$

is a tighter overestimator when $x \in (x_l, 0)$. Thus, the overestimator for $t = x^{1/3}$ is given by the minimum of two lines (A.8) and (A.9).

Further, when $-x_l/8 \geq x_u$, the straight line passing through $(x_l, x_l^{1/3})$ and $(x_u, x_u^{1/3})$, given by

$$t = \frac{(x_u^{1/3} - x_l^{1/3})x + (x_u x_l^{1/3} - x_l x_u^{1/3})}{x_u - x_l}, \tag{A.10}$$

always lies above the curve $x^{1/3}$, which gives the strongest possible concave overestimator. Thus, the unified concave overestimator for $x^{1/3}$ is given by

$$t \leq \begin{cases} \min\left\{\left(x_l^{-2/3} - \frac{2}{3}x_l^{-1}x_u^{1/3}\right)x + \frac{2}{3}x_u^{1/3}, \ \dfrac{x + 2x_u}{3x_u^{2/3}}\right\}, \\ \quad -x_l/8 < x_u, \\[2mm] \dfrac{(x_u^{1/3} - x_l^{1/3})x + (x_u x_l^{1/3} - x_l x_u^{1/3})}{x_u - x_l}, \\ \quad -x_l/8 \geq x_u \end{cases} \tag{A.11}$$

The construction of the concave overestimator for $x^{1/3}$ is graphically illustrated in Fig. 12.

By similar arguments, we can derive the convex underestimator for $x^{1/3}$ when $x_l \leq 0 \leq x_u$ as

$$t \geq \begin{cases} \max\left\{\left(x_u^{-2/3} - \frac{2}{3}x_u^{-1}x_l^{1/3}\right)x + \frac{2}{3}x_l^{1/3}, \ \dfrac{x + 2x_l}{3(-x_l)^{2/3}}\right\}, \\ \quad -x_u/8 > x_l, \\[2mm] \dfrac{(x_u^{1/3} - x_l^{1/3})x + (x_u x_l^{1/3} - x_l x_u^{1/3})}{x_u - x_l}, \\ \quad -x_u/8 \leq x_l. \end{cases} \tag{A.12}$$

The construction of the convex underestimator for $x^{1/3}$ is graphically illustrated in Fig. 13.

Fig. 13 Construction of the convex underestimator for $x^{1/3}$, illustrated with the *solid, black curve*. (**a**) The convex underestimator when $-x_u/8 > x_l$ is given by the maximum of the *dashed red and blue lines*. (**b**) The convex underestimator when $-x_u/8 \leq x_l$ is given by the *dashed blue line*

(a)

(b)
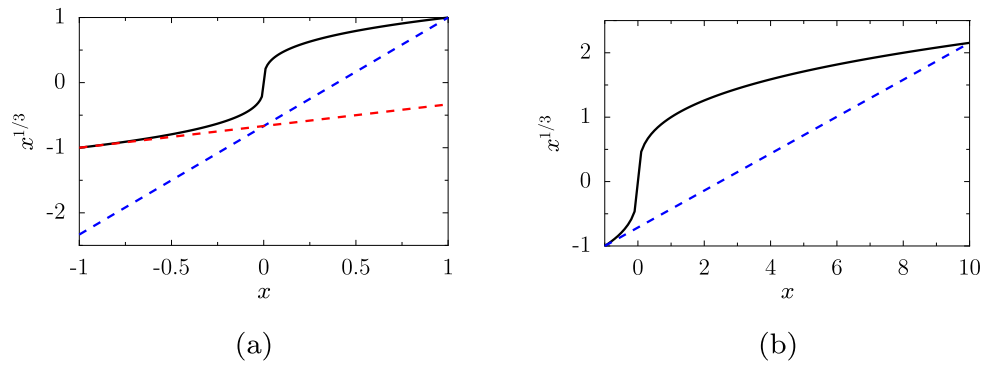
A.4 Convergence Proofs

To be usable in a branch and bound algorithm, the convex relaxations must satisfy the technical conditions (L1) and (L2) described in Sect. 4. Condition (L1) is evidently true by construction. That the Cauchy continuity condition (L2) holds for various bilinear relaxations follows from (McCormick 1976; Al-Khayyal and Falk 1983; Horst and Tuy 2006). We refer the reader to (Tawarmalani and Sahinidis 2001, 2002) to prove the same for Case I in Sect. A.3. Below, we provide a brief proof of the Cauchy continuity condition (L2) for Case II in Sect. A.3.

First, we must prove that as the interval within which x lies becomes smaller, the concave overestimator given by (A.11) becomes tighter. Let us consider the line given by (A.9). The condition (L2) is satisfied when the RHS in (A.9) decreases with an increase in x_l or a decrease in x_u, that is, when $\frac{dt}{dx_l} < 0$ and $\frac{dt}{dx_u} > 0$. Now,

$$\frac{dt}{dx_l} = \frac{2}{3}\frac{x}{x_l^2}(x_u^{1/3} - x_l^{1/3}) \qquad (A.13)$$

which is negative when $x \in [x_l, 0)$, which is the region where (A.9) is active. Further,

$$\frac{dt}{dx_u} = \frac{2}{9}x_u^{-2/3}\left(1 - \frac{x}{x_u}\right) \qquad (A.14)$$

which is non-negative when x lies in the interval $[x_l, x_u]$.

Next, we notice that the part of the overestimator given by (A.8) is independent of the value of x_l, since it is active in the region $[0, x_u]$. In that interval,

$$\frac{dt}{dx_u} = \frac{2}{9}x_u^{-1/3} \qquad (A.15)$$

which is positive since $x_u > 0$.

When $-x_l/8 \geq x_u$, the overestimator given by the line (A.10) must also satisfy the same conditions. Indeed,

$$\frac{dt}{dx_l} = \frac{1}{(x_u - x_l)} \cdot \left[\frac{1}{3}x_l^{-2/3}(x_u - x_l) - \left(x_u^{1/3} - x_l^{1/3}\right)\right]$$

$$= \frac{1}{3(x_u - x_l)} \cdot \left[\left(\frac{x_u}{x_l}\right)^{2/3} + \left(\frac{x_u}{x_l}\right)^{1/3} - 2\right]$$

$$= \frac{1}{3(x_u - x_l)} \cdot \left[\left(\frac{x_u}{x_l}\right)^{1/3} - 1\right] \cdot \left[\left(\frac{x_u}{x_l}\right)^{1/3} + 2\right],$$

where we have used the identity that for any numbers a and b, $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$. Since $x_u - x_l > 0$, it follows that the above expression is negative when

$$-2 < \left(\frac{x_u}{x_l}\right)^{1/3} < 1 \qquad (A.16)$$

or equivalently, when $-8 < \frac{x_u}{x_l} < 1$, which is identically true since the interval under consideration satisfies $-x_l/8 \geq x_u$, $x_l < 0$ and $x_u > 0$.

Finally, it can be shown by similar derivations, or deduced by the symmetry of (A.10), that $\frac{dt}{dx_u} > 0$ when $-x_l/8 \geq x_u$.

By similar arguments as above, it can be shown that the convex underestimator of $x^{1/3}$ given by (A.12) becomes tighter as the size of the interval decreases. This concludes the proof of the Cauchy continuity condition (L2) for Case II in Sect. A.3.

References

Agrawal, M. (2004). On automatic determination of varying focal lengths using semidefinite programming. In *International conference on image processing*.

Agrawal, S., Chandraker, M., Kahl, F., Belongie, S., & Kriegman, D. (2006). Practical global optimization for multiview geometry. In *European conference on computer vision* (pp. 592–605).

Agarwal, S., Snavely, N., & Seitz, S. (2008). Fast algorithms for l_∞ problems in multiview geometry. In *IEEE conference on computer vision and pattern recognition*.

Al-Khayyal, F., & Falk, J. (1983). Jointly constrained biconvex programming. *Mathematics of Operations Research, 8,* 273–286.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Breuel, T. (2002). A comparison of search strategies for geometric branch and bound algorithms. In *European conference on computer vision* (pp. 837–850).

Chandraker, M., Agarwal, S., Kahl, F., Nistér, D., & Kriegman, D. (2007a). Autocalibration via rank-constrained estimation of the absolute quadric. In *IEEE conference on computer vision and pattern recognition*.

Chandraker, M., Agarwal, S., Kriegman, D., & Belongie, S. (2007b). Globally optimal affine and metric upgrades in stratified autocalibration. In *International conference on computer vision*.

Faugeras, O. D. (1992). What can be seen in three dimensions with an uncalibrated stereo rig. In *European conference on computer vision* (pp. 563–578). Berlin: Springer.

Faugeras, O., Luong, Q. T., & Maybank, S. (1992). Camera self-calibration: theory and experiments. In *European conference on computer vision* (pp. 321–334).

Freedman, D. (2003). Effective tracking through tree-search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), 604–615.

Fusiello, A., Benedetti, A., Farenzena, M., & Busti, A. (2004). Globally convergent autocalibration using interval analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12), 1633–1638.

Gat, Y. (2003). A branch-and-bound technique for nano-structure image segmentation. In *Computer vision and pattern recognition workshop*.

Hartley, R. I. (1998). Chirality. *International Journal of Computer Vision*, 26(1), 41–61.

Hartley, R., & Kahl, F. (2007). Optimal algorithms in multiview geometry. In *Asian conference on computer vision* (pp. 13–34).

Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

Hartley, R., Gupta, R., & Chang, T. (1992). Stereo from uncalibrated cameras. In *IEEE conference on computer vision and pattern recognition* (pp. 761–764), Champaign, USA.

Hartley, R. I., Hayman, E., de Agapito, L., & Reid, I. (1999). Camera calibration and the search for infinity. In *International conference on computer vision* (pp. 510–517).

Henrion, D., & Lasserre, J. B. (2003). GloptiPoly: global optimization over polynomials with Matlab and SeDuMi. *ACM Transactions on Mathematical Software*, 29(2), 165–194.

Heyden, A., & Åström, K. (1996). Euclidean reconstruction from constant intrinsic parameters. In *International conference on pattern recognition* (pp. 339–343).

Horst, R., & Tuy, H. (2006). *Global optimization: deterministic approaches*. Berlin: Springer.

Kahl, F. (2005). Multiview geometry and the l_∞-norm. In *International conference on computer vision*.

Kahl, F., & Henrion, D. (2005). Globally optimal estimates for geometric reconstruction problems. In *International conference on computer vision* (pp. 978–985).

Lampert, C., Blaschko, M., & Hofmann, T. (2008). Beyond sliding windows: object localization by efficient subwindow search. In *IEEE conference on computer vision and pattern recognition*.

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3), 497–520.

Lasserre, J. B. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11, 796–817.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2, 164–168.

Manning, R., & Dyer, C. (2001). Metric self calibration from screw-transform manifolds. In *IEEE conference on computer vision and pattern recognition* (pp. 590–597).

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11, 431–441.

McCormick, G. (1976). Computability of global solutions to factorable nonconvex programs. *Mathematical Programming*, 10, 147–175.

Moore, R. E. (1966). *Interval analysis*. New York: Prentice-Hall.

Nistér, D. (2004). Untwisting a projective reconstruction. *International Journal of Computer Vision*, 60(2), 165–183.

Olsson, C., Kahl, F., & Oskarsson, M. (2009). Branch-and-bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 783–794.

Pollefeys, M., & Gool, L. V. (1999). Stratified self-calibration with the modulus constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8), 707–724.

Pollefeys, M., Koch, R., & Van Gool, L. (1998). Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *International conference on computer vision* (pp. 90–95).

Pollefeys, M., Verbiest, F., & Gool, L. J. V. (2002). Surviving dominant planes in uncalibrated structure and motion recovery. In *European conference on computer vision* (pp. 837–851).

Prajna, S., Papachristodoulou, A., & Parrilo, P. (2002). Introducing SOSTOOLS: a general purpose sum of squares programming solver. In *IEEE conference on decision and control*.

Schaffalitzky, F. (2000). Direct solution of modulus constraints. In *Indian conference on computer vision, graphics and image processing* (pp. 314–321).

Sim, K., & Hartley, R. (2006). Recovering camera motion using l_∞ minimization. In *IEEE conference on computer vision and pattern recognition* (pp. 1230–1237).

Stewénius, H., Schaffalitzky, F., & Nistér, D. (2005). How hard is three-view triangulation really? In *International conference on computer vision* (pp. 686–693).

Sturm, J. (1999). Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12, 625–653.

Sturm, P. (2000). A case against Kruppa's equations for camera self-calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), 1199–1204.

Sturm, P., & Triggs, B. (1996). A factorization based algorithm for multi-image projective structure and motion. In *European conference on computer vision* (pp. 709–720).

Tawarmalani, M., & Sahinidis, N. (2001). Semidefinite relaxations of fractional programs via novel convexification techniques. *Journal of Global Optimization*, 20, 137–158.

Tawarmalani, M., & Sahinidis, N. (2002). Convex extensions and envelopes of lower semi-continuous functions. *Mathematical Programming*, 93(2), 247–263.

Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2), 137–154.

Triggs, B. (1997). Autocalibration and the absolute quadric. In *IEEE conference on computer vision and pattern recognition* (pp. 609–614).

Zongker, D., & Jain, A. (1996). Algorithms for feature selection: an evaluation. In *International conference on pattern recognition* (pp. 18–22).