Projective Least-Squares: Global Solutions with Local Optimization

Carl Olsson and Fredrik Kahl
Centre for Mathematical Sciences
Lund University, Sweden

Richard Hartley
Australian National University
and National ICT Australia

Abstract

Recent work in multiple view geometry has focused on obtaining globally optimal solutions at the price of computational time efficiency. On the other hand, traditional bundle adjustment algorithms have been found to provide good solutions even though there may be multiple local minima. In this paper we justify this observation by giving a simple sufficient condition for global optimality that can be used to verify that a solution obtained from any local method is indeed global.

The method is tested on numerous problem instances of both synthetic and real data sets. In the vast majority of cases we are able to verify that the solutions are optimal, in particular for small-scale problems. We also develop a branch and bound procedure that goes beyond verification. In cases where the sufficient condition does not hold, the algorithm returns either of the following two results: (i) a certificate of global optimality for the local solution or (ii) the global solution.

1. Introduction

Multiple view geometry problems are core problems in computer vision and they have been studied for quite some time. The standard way of solving these problems relies on some heuristic method for obtaining an initial estimate, followed by a refinement method that minimizes the sums of squares reprojection error [6]. Due to the local minima problem, global optimization methods have attracted a lot of attention in recent years, see [4] for a survey. Although these methods are guaranteed to compute globally optimal solutions, they are in general slow due to their complexity, e.g., [3, 7, 11]. This makes them less practical since they can not compare with the speed of local, iterative methods such as bundle adjustment [16, 2, 9]. In this paper, we analyse a class of multiview problems that we refer to as *projective least-squares problems* and develop a simple test to verify global optimality. Our experimental results show that local optimization techniques will generally yield a global solution, especially for small-scale problems.

In multiview geometry, an alternative solution to the problem of local minima is to minimize the maximum residual, the so called L_∞-error instead of the L_2-error. It has been shown in [10, 8] that a large number of multiview geometry problems are instances of quasiconvex minimization problems. These problems have no (strict) local minima (other than the global one) and can be solved efficiently using local optimization methods [1]. While the L_∞-norm of reprojection errors is a geometrically meaningful criterion, it is still desirable to find the least squares solution (that is, the L_2-solution) since it is statistically optimal. Under the assumption of independent, Gaussian noise the L_2-solution gives the maximum likelihood estimate.

Although practically useful methods that guarantee global solutions are rare - the factorization algorithm [15] for affine cameras being one exception - it has been empirically found that local methods usually work well with proper initialization [6, 2, 14]. In this paper we present a framework that allows us to verify that an obtained local solution is indeed global. The verification idea was first introduced in [5] for the triangulation problem with spherical cameras, and it was also suggested that the approach might work for other problems. Here we generalize this approach in several directions. Our main contributions are:

- Our framework deals with a whole class of multiview geometry problems and we give conditions that are provably *stronger* than the ones given in [5] for triangulation.
- We do not stop at verification of optimality. It is shown how the verification test can be embedded in a branch and bound algorithm that always gives an answer. Either (i) a certificate of optimality for the local solution or (ii) the global solution is returned.
- We perform an extensive set of experiments on several data sets to explore the limitations of the method. Results are given for *triangulation*, *homography estimation*, *camera resectioning* and *structure and motion with known rotations*. Our main conclusion is that in most cases the verification test succeeds, even for problems with many cameras and several hundred of point features involved.

2. Projective Least-Squares

2.1. Bounding the Location of the L_2-Optimum

Let the functions $f_i(x)$ represent the squared error residuals when the L_2 error cost is considered, that is,

$$f(x) = \sum_{i=1}^{m} f_i(x). \tag{1}$$

In [10, 8], it was shown that the error residuals $f_i(x)$ for a large number of multiview geometry problems are quasiconvex and this is the class to which our analysis is restricted. The property of quasiconvexity guarantees that any (strict) local minimum is also a global minimum. However since the sum of quasiconvex functions is not necessarily quasiconvex the L_2-error (1) may have multiple local minima.

The goal of this paper is to determine a region in which the the global minimum must lie and to show that the L_2-error function is in fact convex in this region. Now suppose we have obtained (through local optimization) a local minimum $f(x^{local}) = \epsilon_{max}^2$ with a minimizer x^{local}. As noted in [5], clearly the global minimum must lie in the region

$$\mathcal{R} = \{\, x \in \mathbb{R}^n; f_i(x) \leq \epsilon_{max}^2 \text{ for } i = 1, \ldots, m \,\}, \tag{2}$$

since each residual must be smaller than the sum of all the residuals. Now since the functions f_i are quasiconvex the region $\mathcal{R}$ is a convex region which must contain the global minimum. Our goal is to reject the hypothesis that there exists another local minimum, different from x^{local} with a lower objective value. Hence if we can show that $f(x)$ is convex on $\mathcal{R}$ we know that our local minimum is in fact global.

2.2. The Hessian of the Perspective Error Function

Since a differentiable function is (strictly) convex if and only if its Hessian is positive definite we will now turn our attention to the Hessian of the perspective function. If (X, Y, Z) are the coordinates of a point in 3D-space then we define the perspective error function as

$$\pi(X, Y, Z) = \frac{X^2 + Y^2}{Z^2}. \tag{3}$$

The mapping $\pi : \{(X, Y, Z) \in \mathbb{R}^3; Z > 0\} \mapsto \mathbb{R}^2$ is the squared distance from the image center to the perspective projection of a 3D-point onto the image plane. The constraint $Z > 0$ reflects the fact that 3D-points should be located in front of the camera.

In this class of so called projective least-squares problems, the error residuals in (1) can always be related to (3) by an affine change of coordinates

$$\begin{aligned} X_i &= a_i^T x + \tilde{a}_i & (4)\\ Y_i &= b_i^T x + \tilde{b}_i & (5)\\ Z_i &= c_i^T x + \tilde{c}_i & (6)\\ f_i(x) &= \pi(X_i, Y_i, Z_i), & (7) \end{aligned}$$

where $x \in \mathbb{R}^n$. Note that the dimensionality of x depends on the particular application, e.g., $n = 3$ in triangulation, $n = 8$ in homography estimation, $n = 11$ in camera resectioning etc.

The Hessian $\nabla^2\pi(X, Y, Z)$ of the perspective error function can be written as

$$\nabla^2\pi(X, Y, Z) = \frac{2}{Z^2} \begin{pmatrix} 1 & 0 & -2\frac{X}{Z} \\ 0 & 1 & -2\frac{Y}{Z} \\ -2\frac{X}{Z} & -2\frac{Y}{Z} & 3\frac{(X^2+Y^2)}{Z^2} \end{pmatrix}. \tag{8}$$

To simplify the notation later on, we introduce the parameters $d = Z$ and $\epsilon = \frac{\sqrt{X^2+Y^2}}{Z}$. Here d is the depth and ϵ is the distance from the image center to the projection of (X, Y, Z). The matrix $\nabla^2\pi$ can be difficult to handle since it is not diagonal. However for our purposes we shall see that it is enough to use a lower bounding matrix D such that $\nabla^2\pi \succeq D$. It was shown in [5] that the diagonal matrix

$$D(X, Y, Z) = \frac{2}{d^2} \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & -3\epsilon^2 \end{pmatrix} \tag{9}$$

can be used to bound $\nabla^2\pi(X, Y, Z)$. Indeed it is easy to verify that the eigenvalues of $\nabla^2\pi - D$ are non-negative, and hence $\nabla^2\pi - D \succeq 0$.

For a general error residual f_i we apply the coordinate change (4)-(7) to obtain

$$f_i(x) = \frac{(a_i^T x + \tilde{a}_i)^2 + (b_i^T x + \tilde{b}_i)^2}{(c_i^T x + \tilde{c}_i)^2}. \tag{10}$$

Using the chain rule (twice) we may compute the Hessian $\nabla^2 f_i(x)$. Since the coordinate change is affine, the expression for the Hessian takes on a particularly pleasing form. Let W_i be the $n \times 3$ matrix with columns a_i, b_i, c_i, then

$$\nabla^2 f_i(x) = W_i \nabla^2\pi(X_i, Y_i, Z_i) W_i^T \tag{11}$$

where X_i, Y_i and Z_i depends on x through (4)-(7). Note that W_i is independent of x. Now if $\nabla^2\pi - D \succeq 0$ holds, then it follows that the same must be true for $W_i(\nabla^2\pi(X_i, Y_i, Z_i) - D(X_i, Y_i, Z_i))W_i^T \succeq 0$. Denote $D_i(x) = D(X_i, Y_i, Z_i)$. Hence the matrix $W_i D_i(x) W_i^T$ can be used as a lower bound on $\nabla^2 f_i(x)$, that is,

$$\nabla^2 f_i(x) \succeq W_i D_i(x) W_i^T. \tag{12}$$

The main point is that if $W_i D_i(x) W_i^T$ is positive semidefinite, then the Hessian $\nabla^2 f_i(x)$ is likewise. As $D_i(x)$ is a diagonal matrix, we can also write

$$W_i D_i(x) W_i^T = \frac{2}{3d_i(x)^2}\left(a_i a_i^T + b_i b_i^T - 9\epsilon_i(x)^2 c_i c_i^T\right). \tag{13}$$

Note that d_i and ϵ_i depends on x through (4) – (6).

2.3. Sufficient Conditions for Convexity

Next we will develop conditions that ensure convexity of the L_2-error function in the domain $\mathcal{R}$ in (2).

Recall that a function is convex if its Hessian is positive semidefinite over its domain. The Hessian of $f(x)$ in (1) can be written

$$\nabla^2 f(x) = \sum_i \nabla^2 f_i(x). \tag{14}$$

Hence the goal is to show that this sum is positive semidefinite. From the previous section we know that $\nabla^2 f_i(x) \succeq W_i D_i(x) W_i^T$ and the same holds true for the sum

$$\sum_i \nabla^2 f_i(x) \succeq \sum_i W_i D_i(x) W_i^T. \tag{15}$$

Therefore, to ensure convexity, it is enough to show that the right hand side of (15) is positive semidefinite in the region $\mathcal{R} \subset \mathbb{R}^n$. Inserting (13) into the right hand side of (15) we obtain

$$\sum_i \frac{2}{3d_i(x)^2}\left(a_i a_i^T + b_i b_i^T - 9\epsilon_i(x)^2 c_i c_i^T\right) \succeq 0. \tag{16}$$

According to the definition of positive semidefinite the following should hold

$$y^T \sum_i \left(\frac{a_i a_i^T + b_i b_i^T}{d_i(x)^2} - 9\epsilon_i(x)^2 \frac{c_i c_i^T}{d_i(x)^2}\right) y \geq 0, \tag{17}$$

for all $y \neq 0$ and all $x \in \mathcal{R} \subset \mathbb{R}^n$. Now for a fixed y, the terms $y^T a_i a_i^T y = (a_i^T y)^2$ and $y^T b_i b_i^T y = (b_i^T y)^2$ are constant and non-negative over the domain. Hence $y^T(\frac{a_i a_i^T + b_i b_i^T}{d_i(x)^2})y$ attains its minimum for the largest feasible depth $d_{i,max}$. Such bounds for $d_i(x)$, specifically

$$d_{i,min} \leq d_i(x) \leq d_{i,max}, \tag{18}$$

may be computed by solving two simple linear programs (or SOCPs depending on the choice of residuals) over $\mathcal{R}$, see Section 3.1. Therefore the terms in (17) can be bounded by

$$y^T(\frac{a_i a_i^T + b_i b_i^T}{d_i(x)^2})y \quad \leq \quad y^T(\frac{a_i a_i^T + b_i b_i^T}{d_{i,min}^2})y, \tag{19}$$

$$y^T \frac{c_i c_i^T}{d_i(x)^2} y \quad \geq \quad y^T \frac{c_i c_i^T}{d_{i,max}^2} y. \tag{20}$$

Furthermore, since $\epsilon_i(x) \leq \epsilon_{max}$, it follows that if

$$y^T \sum_i \left(\frac{a_i a_i^T + b_i b_i^T}{d_{i,max}^2} - 9\epsilon_{max}^2 \frac{c_i c_i^T}{d_{i,min}^2}\right) y \geq 0 \tag{21}$$

holds for all y, then the function is convex in the domain of interest. Finally, it is a well known fact that, if we restrict y to $||y|| = 1$, then the minimum of the left hand side is equal to the smallest eigenvalue of the matrix, and therefore it is sufficient to test if

$$\lambda_{min}\left(\sum_{i=1}^m \left(\frac{a_i a_i^T + b_i b_i^T}{d_{i,max}^2} - 9\epsilon_{max}^2 \frac{c_i c_i^T}{d_{i,min}^2}\right)\right) \geq 0. \tag{22}$$

2.4. Comparison to the Triangulation Bounds of [5]

In [5], Hartley and Seo studied the problem of triangulation with calibrated cameras. One may assume that the camera matrices are of the form $P_i = [R_i\, t_i]$ where R_i is a rotation matrix. Hence the coordinate change (4)-(6) is given by

$$(a_i^T, \tilde{a}_i) = (R_i^1, t_i^1) \tag{23}$$

$$(b_i^T, \tilde{b}_i) = (R_i^2, t_i^2) \tag{24}$$

$$(c_i^T, \tilde{c}_i) = (R_i^3, t_i^3) \tag{25}$$

where R_i^j denotes row j of R_i and t_i^j is the j'th coordinate of t_i. The bound which was obtained in [5] can be written in our notation as

$$\lambda_{min}\left(\sum_i \frac{a_i a_i^T + b_i b_i^T}{d_{i,max}^2}\right) \geq 9\epsilon_{max}^2 || \sum_i \frac{c_i c_i^T}{d_{i,min}^2} ||, \tag{26}$$

using the relation $I = a_i a_i^T + b_i b_i^T + c_i c_i^T$. The norm on the right hand side is the same as the maximum eigenvalue of $\sum_i \frac{c_i c_i^T}{d_{i,min}^2}$. Now it is easy to see that unless the eigenvectors corresponding to the minimal eigenvalue on the left hand side also happen to be eigenvectors corresponding to the maximum eigenvalue of $\sum_i \frac{c_i c_i^T}{d_{i,min}^2}$, (22) is a strictly stronger bound than (26). That is, (26) implies (22) but not the other way around. In [5], a weighting function is also introduced to improve the bound in (26). However, such weighting functions have no effect on our bound. It is easy to show that the test will give the same answer regardless of any weighting.

The most important difference to [5] is, however, not that our bound is stronger, but that (i) we can handle a larger class of problems and (ii) we are able to go beyond verification, see Section 4.

3. Verifying Optimality

3.1. Bounding the Depths

Each residual $f_i(x)$ in (10) satisfies $f_i(x) \leq \epsilon_{max}^2$. Since each $f_i(x)$ is a quasiconvex function, the constraint can be rewritten as a convex constraint. The maximum and minimum depths over $\mathcal{R}$ may be computed by solving the following convex programs for $i = 1, ..., m$,

$$\max \text{ or } \min c_i^T x + \tilde{c}_i$$
$$||a_j^T x + \tilde{a}_j, b_j^T x + \tilde{b}_j|| \leq \epsilon_{max}(c_j^T x + \tilde{c}_j), \quad j = 1, ..., m.$$

This type of program is a so called Second Order Cone Program (SOCP) and it is easily solvable using commonly available software packages, for example, SeDuMi [13].

Slightly looser bounds on the depths can be obtained by instead solving the following linear programs for $i = 1, ..., m$,

$$\begin{array}{rcl} \max & \text{or} & \min \quad c_i^T x + \tilde{c}_i \\ |a_j^T x + \tilde{a}_j| & \leq & \epsilon_{max}(c_j^T x + \tilde{c}_j) \\ |b_j^T x + \tilde{b}_j| & \leq & \epsilon_{max}(c_j^T x + \tilde{c}_j), \quad j = 1, ..., m. \end{array}$$

Linear programming is considerably faster than SOCP. For our experiments, we have tried both methods of bounding the depths but we have not found any significant difference between the resulting depths.

3.2. A Further Improved Sufficient Condition

If the minimum and maximum depths differ a lot the condition derived in (22) may become weak, and insufficient to prove convexity, even though the original problem is convex. We will take a step back and derive an improved procedure for verifying optimality.

Consider again the sufficient condition in (16). As before, we may replace $\epsilon(x)$ by ϵ_{max} to obtain another sufficient condition. This condition is equivalent to verifying that

$$\sum_{i=1}^{m} \mu_i \left(a_i a_i^T + b_i b_i^T - 9\epsilon_{max}^2 c_i c_i^T \right) \succeq 0, \qquad (27)$$

is fulfilled for $\frac{1}{d_{i,max}^2} \leq \mu_i \leq \frac{1}{d_{i,min}^2}$, $i = 1, \ldots, m$. The bounds on μ_i determine a hyperrectangle in $\mathbb{R}^m$. If condition (27) is fulfilled when evaluated at all extreme points (corners) of this hyperrectangle, it follows by convexity that the condition is fulfilled for the whole domain $\frac{1}{d_{i,max}^2} \leq \mu_i \leq \frac{1}{d_{i,min}^2}$, $i = 1, \ldots, m$.

This condition is stronger than (22) in the sense that if (22) is satisfied then (27) will automatically be satisfied, but the other way around might not be true. However, there is a price to pay. There are 2^m extreme points to check and if the number of residuals m is large then this is simply intractable.

3.3. Projective Coordinate Change

By changing projective coordinate systems, we will reparametrize the domain of the error function $f(x)$. This may turn a non-convex function into a convex function. It is easy to see that affine coordinate changes do not affect convexity so it is enough to restrict our attention to transformations that change the plane at infinity. Cheirality conditions also apply, that is, all 3D-points must remain in front of the cameras. See [6] for more details on projective geometry.

Even if there exists a transformation $T : \mathbb{P}^n \mapsto \mathbb{P}^n$ which makes the error function $f(x)$ convex over the transformed domain $T(\mathcal{R})$, it is a hard optimization problem to find it. Note that the maximum and minimum depths change when the plane at infinity is transformed, so it is required that the bounding depths are recomputed as well. We have tried different strategies and have found that maximizing λ_{min} in (22) directly over the set of (feasible) planes at infinity works well. As the objective function is not differentiable, random search directions are applied to the current solution in order to iteratively find better and better solutions. If the minimum eigenvalue becomes non-negative, then the convexity condition holds.

3.4. Test Procedure

The following steps are performed in order to verify optimality:

1. Compute a local minimizer x^{local} (with bundle adjustment).

2. Compute maximum/minimum depths over $\mathcal{R}$ in (2), see next section for details.

3. Test if the convexity condition in (22) holds.

For most problem instances (see experimental section), this test will give a positive answer and we are done. If the condition is not fulfilled, then we cannot draw any definite conclusion. Still, there is more we can do.

4. Test a stronger convexity condition, see Section 3.2.

If this test also fails, then again, one cannot draw any conclusion. Either x^{local} is not the global minimizer or the sufficient conditions are simply too weak. In Section 4 we will show how one can determine which is the correct conclusion using a more sophisticated procedure. The price to pay is computational effort.

5. Change projective coordinate systems and test condition (22) again, see Section 3.3.

Convexity is *not* invariant under projective transformations, so by changing coordinate systems it may in some cases be possible to verify convexity.

4. Computing the Global Solution

When the procedure for verifying optimality fails, it is a bit unsatisfactory that the reason for failure is unclear. Either there exists a better local minimum or the sufficient condition is too weak. In this section we will devise an algorithm that will always give an answer to the problem at hand.

The strength of condition (22) depends on the size ϵ_{max} of the region $\mathcal{R}$ defined in Section 2.1. For large scale reconstructions, with potentially thousands of residuals, it is possible that ϵ_{max} may become large, and the error function $f(x)$ is non-convex.

The idea is to reduce the feasible domain $\mathcal{R}$ into two smaller domains $\mathcal{R}_1$ and $\mathcal{R}_2$ and then try to verify optimality in each domain separately. We may also find other (local) minimizers in the subdomains which can turn out to be the actual global solution, denoted x^{global}, we are looking for. The process is iterated until convergence.

Let $i \in \{1, \dots, m\}$. Then, for every $\eta \in (0,1)$, exactly one of the following two hypotheses will be true:

1. $f_i(x^{global}) \in [\,0, \eta\epsilon^2_{max}\,]$.
2. $f_i(x^{global}) \in (\,\eta\epsilon^2_{max}, \epsilon^2_{max}\,]$, which implies that $f_j \leq (1-\eta)\epsilon^2_{max}$ for all $j \in \{\,1, \dots, m\,\} \setminus \{i\}$.

The two hypotheses will generate two convex regions $\mathcal{R}_1 \subset \mathbb{R}^n$ and $\mathcal{R}_2 \subset \mathbb{R}^n$ whose union will be smaller than the original domain, that is, $\mathcal{R}_1 \cup \mathcal{R}_2 \subset \mathcal{R}$. In order to prove optimality, it is sufficient to verify optimality for the two regions $\mathcal{R}_1$ and $\mathcal{R}_2$.

The branching of regions into smaller regions will generate a tree of nodes. Each residual $f_i(x)$ in every node will have a lower and an upper bound. For each node, the test procedure described in Section 3.4 is applied. Of course, if the minimizer x^{local} is already in the domain, there is no need to recompute it. The following things can happen:

- The feasible region is empty. This node in the tree needs no further examination.
- The sum of upper bounds of squared residual error is lower than ϵ^2_{max}. If the domain is non-empty, one can conclude there is a lower-residual solution compared to $f(x^{local}) = \epsilon^2_{max}$.
- The sum of lower bounds of squared residual error is higher than ϵ^2_{max}. Such domains cannot contain the global solution and can be discarded.
- The convexity test holds and hence the L_2 error function is convex on the domain. One can conclude that there is only one possible local minimum. The node needs no further examination.
- The convexity test fails. The node needs to be branched upon again and put in the list of nodes to examine.

This is a classical branch and bound scheme. There are many possible branching strategies. We simply choose to branch on the residual with the largest interval length and we set $\eta = 1/2$ in the experiments. Convergence follows from the fact that there always exists an open set containing the global minimum where the function is convex.

5. Experiments

Synthetic Data. To test the sensitivity to noise of the verification test (steps 1-3 in Section 3.4), we artificially generated a number of synthetic instances of triangulation, homography estimation and camera resectioning (i.e., uncalibrated camera pose) which are projective least-squares problems. The data was generated by randomly placing 3D-points inside the unit cube of $\mathbb{R}^3$ and the cameras were randomly placed on the sphere of radius 2 directed at the origin. For homography estimation, the 3D-points were placed on the plane $z = 0$ and the cameras were located on the same side of the plane. The size of the obtained images is around 2×2 units. Finally we added noise with (varying) standard deviation σ to the image measurements.

Figure 1 shows the fraction of problems that could be verified to be optimal as a function of σ, using the basic bound (22). Each local minimizer was computed with bundle adjustment. For each noise level we ran 250 instances of the problem. As can be seen, we are able to verify optimality for much higher noise levels in the triangulation case. One likely reason is that for similar noise levels, the sum of squared residuals ϵ^2_{max} will be larger for resectioning and homography estimation compared to triangulation since we have a larger number of residuals. For triangulation, bundle adjustment finds verifiable solutions for noise levels up to 0.05 units (which corresponds to measurement errors of 2.5% of the image size), whereas for resectioning and homography estimation, the corresponding number is 0.005 pixels (0.25%).

Note that it is not possible to decide whether there is a local minimum (with residual error less than ϵ^2_{max}) if the verification test is unsuccessful. All we can say is that we are certain that there is none if the test is successful.

Notre Dame. This data set was originally created in [12] and it was also used to verify optimality for triangulation by Hartley and Seo in [5]. We perform an experimental comparison of the bounds derived in this paper (cf. Section 2.4).

Out of 277887 triangulated 3D-points, 229 instances could not be verified with the bound of Hartley and Seo (26). With our primary bound in (22), 224 instances were unsuccessful and after applying the stronger secondary

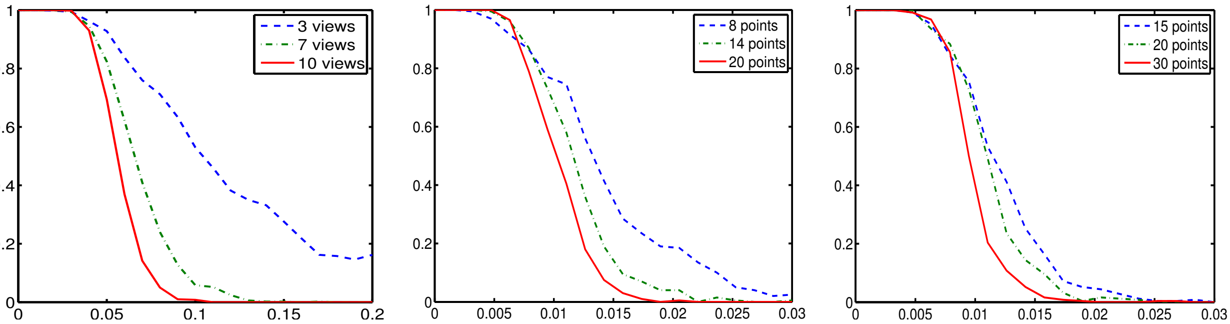


Figure 1. Synthetic experiments. Fraction of solutions that could be verified vs. the standard deviation of the noise level. Left: triangulation, middle: homography estimation, right: resectioning. (A noise level of 0.01 corresponds to a pixel error of 0.5 % of the image.)

bound derived in Section 3.2, 174 cases remained. By transforming the plane at infinity, another 134 cases could be proven to be optimal. The remaining 40 triangulation problems were proven to be global solutions by the branch and bound method in Section 4. Only a few branching steps were required to provide the certificate of optimality. Note that all 277887 problem instances were proven to be globally optimal and not a single local optimum was found! See Table 1 for a summary of the results. Each problem instance takes on the average less than 2.0 milliseconds to verify on a 3.2GHz Pentium written in C++ (excluding the time for branch and bound).

In the case of triangulation, the practical difference between the bound by Hartley and Seo compared to our primary bound is minor. However, when adapting the bound of Hartley and Seo to other problems with more degrees of freedom, we have found that the difference is significant. For example, almost all the instances of structure and motion with known rotation in Tables 1-5 were unsuccessful.

Dinosaur. This turn-table sequence consists of 36 images with 328 given point features. The complete 3D reconstruction was computed with standard structure from motion routines (including refinements by bundle adjustment).

The multiview geometry problems tested were triangulation, camera resectioning and structure and motion with known rotations. See Table 2 for a summary of results. Both camera resectioning and triangulation work very well for this type of scene and camera motion. It was not possible to prove optimality for the whole sequence (assuming known rotations), only for up to 22 images. Note that this setup is still a large structure and motion problem: 22 cameras and several hundreds of 3D points. When going beyond 22 images, the primary bound was not sufficient. We did not apply the secondary bound, projective transformations nor branch and bound due to the size of the setup. We also tried verifying 100 random pairs of images with common feature points in this pair, and similarly 100 random triples with all common feature points, with 100% success rate. Only pairs and triples with at least 10 feature points were selected.

The Christ Statue. The images were collected from various tourist photographs of this well-known statue and reconstructed with standard structure and motion routines. Similar experiments as for the Dinosaur sequences were performed with similar success rates, see Table 3.

Corridor. The forward camera motion for this 11-image sequence is quite different from the other sequences. Verifying global optimality for triangulation and camera resectioning turned out to be no problem, but for the other multiview problems it was more difficult, see Table 4.

There are several 3D planes in the scene and we took all the point features on the left frontal wall and computed all pairwise 3×3 homographies for this plane using bundle adjustment. Out of the $\binom{11}{2} = 55$ pairs, 27 were successfully verified with the primary bound. The secondary bound was not applied since there are way too many residuals (around 100 correspondences). Using projective coordinate changes, we were able to verify all but 6 homographies. The 6 remaining pairs were verified using the branch and bound with only one branching step in each case.

The known rotation case turned out to be more difficult, which is consistent with the findings in [17]. Not a single test was successful for the primary bound. We did not apply the other methods due to the size of the setup. The whole sequence has 11 cameras and more than 4000 image points, so just computing all the depth bounds takes time. (The size for the LP constraint set is approximately $4 \times 4000 = 16000$ since each image coordinate gives rise to two inequality constraints, see Section 3.1).

Arnold. A final experiment with 6 images of a poster was tested, see Figure 2 and Table 5. In this scene, all 3D feature points are on a plane, and hence uncalibrated camera resectioning is impossible. Triangulation of all 3D points (in total, 1639 points) as well as all pairwise homographies (15 in total) induced by the plane were successfully verified by the primary bound. The results with known rotation were as follows. 9 out of 15 pairwise images were successfully verified and 17 out 20 image triples were verified.

To summarize, reconstruction problems with small dimensions can always be handled with little extra computational effort (the primary bound generally works) while for larger problems the picture is more mixed. The success rate depends on the geometry of the scene as well as the motion of the camera. Images taken with a wide baseline seem to be more easily verifiable compared to other camera motions. As the number of residuals increases, it becomes harder to verify optimality - just as expected.

6. Conclusions

We have shown how to compute global solutions for multiple view geometry problems. The goal of getting globally optimal solutions in the maximum likelihood sense is actually tractable even for large scale problems. From a practical point of view, one can conclude that it is relatively inexpensive (in terms of computational effort) to guarantee optimality for problems with small dimensions (like triangulation and camera resectioning) while for larger problems the picture is more mixed. The experiments indicate that it is relatively cheap to verify optimality for problems with large baselines.

Another conclusion is that for projective least-squares problems, the local minima problem does not seem to be a major problem. However, we should bear in mind that our experiments are biased since they are based on sequences that have been successfully reconstructed with standard structure and motion methods. So, bad solutions have already been eliminated with (good) heuristics. With the tools proposed in this paper, one could identify if a bad solution was due to the local minima problem or if it was caused by some other problem (for example, outliers).

Acknowledgments. This work has been funded by the European Research Council (GlobalVision grant no. 209480), the Swedish Research Council (grant no. 2007-6476) and the Swedish Foundation for Strategic Research (SSF) through the programme Future Research Leaders.

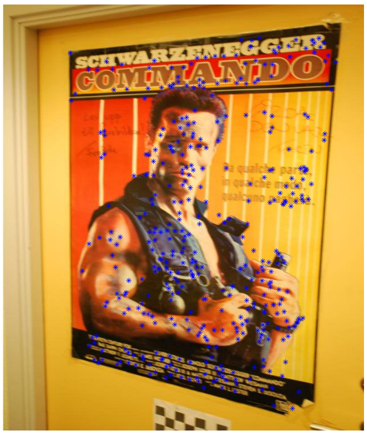


Figure 2. One image of the Arnold sequence with feature points.

References

[1] S. Agarwal, N. Snavely, and S. Seitz. Fast algorithms for L_∞ problems in multiview geometry. In *Conf. Computer Vision and Pattern Recognition*, Anchorage, USA, 2008.

[2] C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In *Photogrammetric Computer Vision (PCV)*, 2006.

[3] A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using interval analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(12):1633–1638, 2004.

[4] R. Hartley and F. Kahl. Optimal algorithms in multiview geometry. In *ACCV*, pages 13–34, 2007.

[5] R. Hartley and Y. Seo. Verifying global minima for L_2 minimization problems. In *Conf. Computer Vision and Pattern Recognition*, Anchorage, USA, 2008.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[7] F. Kahl, S. Agarwal, M. K. Chandraker, D. J. Kriegman, and S. Belongie. Practical global optimization for multiview geometry. *Int. Journal Computer Vision*, 79(3):271–284, 2008.

[8] F. Kahl and R. Hartley. Multiple view geometry under the L_∞-norm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008.

[9] N. Kai, D. Steedly, and F. Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. In *Conf. Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007.

[10] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1834–1847, 2007.

[11] C. Olsson, F. Kahl, and M. Oskarsson. Branch and bound methods for Euclidean registration problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009.

[12] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM SIGGRAPH*, 25(3):835–846, 2006.

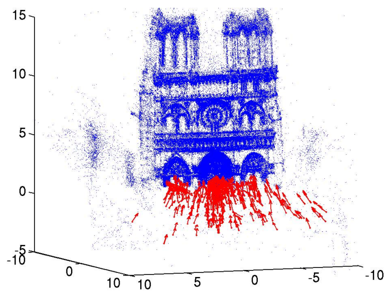
[13] J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[14] J.-P. Tardif, A. Bartoli, M. Trudeau, N. Guilbert, and S. Roy. Algorithms for batch matrix factorization with application to structure-from-motion. In *Conf. Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007.

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

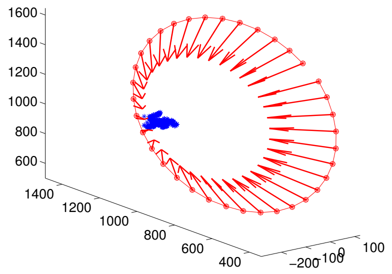
[16] B. Triggs, P. McLauchlan, H. R.I, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms'99*, pages 298–372, in conjunction with ICCV'99, Kerkyra, Greece, 1999.

[17] A. Vedaldi, G. Guidi, and S. Soatto. Moving forward in structure from motion. In *Conf. Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007.



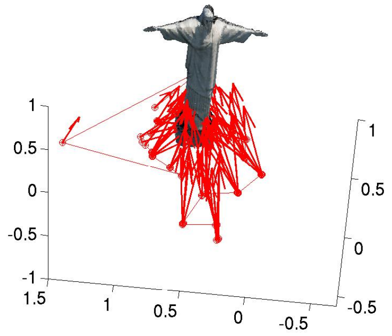
Problem	Instances	Method	Unsuccessful tests
Triangulation	277887 points	Hartley and Seo (26)	229
		Primary bound (22)	224
		Secondary bound (27)	174
		Projective change	40
		Branch and bound	0

Table 1: Results for Notre Dame.



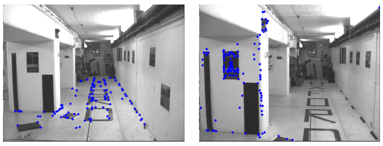
Problem	Instances	Method	Unsuccessful tests
Triangulation	328 points	Primary bound (22)	0
Resectioning	36 cameras	Primary bound (22)	0
Known rotation	36 images	Primary bound (22)	1
	22 images	Primary bound (22)	0
	100 random pairs	Primary bound (22)	0
	100 random triples	Primary bound (22)	0

Table 2: Results for Dinosaur.



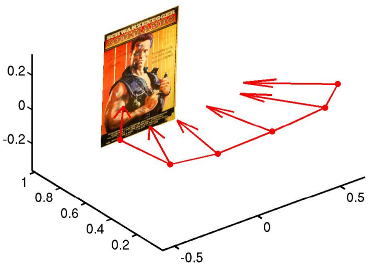
Problem	Instances	Method	Unsuccessful tests
Triangulation	185 points	Primary bound (22)	0
Resectioning	76 cameras	Primary bound (22)	0
Known rotation	100 random pairs	Primary bound (22)	12
		Projective change	11
	100 random triples	Primary bound (22)	3
		Projective change	3

Table 3: Results for the Christ Statue.



Problem	Instances	Method	Unsuccessful tests
Triangulation	737 points	Primary bound (22)	0
Resectioning	11 cameras	Primary bound (22)	0
Homography	55 pairs	Primary bound (22)	28
		Projective change	6
		Branch and bound	0
Known rotation	11 images	Primary bound (22)	1
	55 pairs	Primary bound (22)	55
	165 triples	Primary bound (22)	165

Table 4: Results for Corridor.



Problem	Instances	Method	Unsuccessful tests
Triangulation	1639 points	Primary bound (22)	0
Homography	15 pairs	Primary bound (22)	0
Known rotation	15 pairs	Primary bound (22)	6
	20 triples	Primary bound (22)	3

Table 5: Results for Arnold.