Multiple View Geometry and the L_∞-norm

Fredrik Kahl
fredrik@maths.lth.se

Computer Science and Engineering
University of California San Diego, USA

Centre for Mathematical Sciences
Lund University, Sweden

Abstract

This paper presents a new framework for solving geometric structure and motion problems based on L_∞-norm. Instead of using the common sum-of-squares cost-function, that is, the L_2-norm, the model-fitting errors are measured using the L_∞-norm. Unlike traditional methods based on L_2, our framework allows for efficient computation of global estimates. We show that a variety of structure and motion problems, for example, triangulation, camera resectioning and homography estimation can be recast as a quasi-convex optimization problem within this framework. These problems can be efficiently solved using Second Order Cone Programming (SOCP) which is a standard technique in convex optimization. The proposed solutions have been validated on real data in different settings with small and large dimensions and with excellent performance.

1. Introduction

Structure and motion problems form a class of geometric reconstruction problems where the goal is to infer the scene structure (often 3D points) and/or the camera motion, given image data. Let P_i, $i = 1, ..., m$, denote a set of 3×4 camera matrices, U_j, $j = 1, ..., n$, a set of 3D points, and u_{ij} the projected image points, all represented in homogeneous coordinates. So, given u_{ij}, the objective is to recover P_i and/or U_j under a pinhole camera model $u_{ij} \simeq P_i U_j$.

The correct procedure for reconstructing the unknowns (again, either structure or motion, and in some cases both) is to find the solution which reproduces the images as closely as possible. In other words, we want to minimize the geometric distances between the measured image points and the reprojected structure and motion parameters. Let $d(\cdot, \cdot)$ denote the geometric distance between two image points and let r be the residual vector $r = [..., d(u_{ij}, P_i U_j), ...]^T$ containing all such mn distances. Whence, one is led to the following optimization problem: $\min ||r||$, where $|| \cdot ||$ is usually the L_2-norm. The purpose of this paper is to investigate what simplifications can be obtained if the L_2-norm is replaced by the L_∞-norm, thus analyzing: $\min ||r||_\infty$.

The main contribution of this paper is the introduction of a L_∞-framework which allows for efficient computations of global estimates for a wide class of geometric vision problems. The solutions are invariant with respect to projective transformations of the world coordinate system and to similarity transformations in the image plane, which follows from the well-known fact that the image metric $d(\cdot, \cdot)$ itself is invariant to such transformations. Therefore, there is no need for normalization of the image coordinates which is prerequisite for all algebraic methods. Another important contribution is the introduction of the optimization framework of second order cone programming (SOCP) and how SOCP can be applied to reconstruction problems involving rational polynomials. The technique may have applications in other areas of computer vision.

Solving for structure and motion with L_2-norm is a hard non-convex problem. Globally optimal estimates can only be computed for rare instances of the problem. For example, a solution to the triangulation problem for two views was given in [4] for L_2-norm and in [10] for L_∞-norm. Another important example is the factorization algorithm [14], but it is limited to the affine camera model. The projective generalizations of the factorization approach do not generally optimize the L_2-norm, cf. [13, 6]. Although there are recent and promising attempts of computing global estimates of non-convex problems [2, 7], they are limited to problems of small dimensions that are computationally demanding and rather cumbersome to implement.

In general, one has to rely on local, iterative techniques, so-called bundle adjustment methods [16]. In turn, such methods are reliant on good initialization in order to avoid local minima. However, the initialization techniques frequently used, e.g., the eight-point algorithm [9, 15], optimize some algebraic cost-function which simplifies the problem, but it has no geometrical or statistical meaning. When significant measurement noise is present, such estimates may be far from the global optimum.

L_∞-optimization can be regarded as something in between the statistically optimal L_2-methods and linear algorithms. The L_∞-framework inherits good properties from both of these alternative approaches. For example, global estimates are guaranteed with a geometrically meaningful cost-function and at a reasonable computational cost. A potential

disadvantage is that the L_∞-norm is not robust to outliers. The ultimate test for this is to try the method on real data and evaluate its performance.

The most closely related work is [3] and it was an inspiration for the present paper. It was shown that the L_∞-triangulation problem for an arbitrary number of views can have only one (local) minimum and hence it is the global minimum. Then, the problem of reconstructing 3D points and camera centres using an angular image norm was investigated. An algorithm based on line search was described, but it did not work well for problems with many degrees of freedom. Also, no experimental results were given. We generalize their work in several directions. First of all, a class of geometric reconstruction problems is handled within our framework using the L_∞-norm of the reprojection errors. At all times, the solutions are invariant with respect to coordinate transformations of the image and the world. And perhaps most importantly, an efficient algorithm is presented based on standard convex optimization techniques, capable of handling large-scale problems.

This paper is organized as follows. In the next section, an introduction to the necessary machinery of convex optimization is introduced. Then, the framework of SOCP and quasiconvex functions is applied to the triangulation problem (Section 3), the estimation of homographies and camera pose (Section 4) and multiview reconstruction problems in Section 5. Finally, experimental results are given in Section 6 followed by a concluding discussion.

2. Convex Optimization

In this section, some notations and concepts of convex optimization are presented. For more details, the reader is referred to the excellent book [1] or [8] for SOCP problems.

A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is *convex* if its domain, $\mathrm{dom} f$, is a convex set and for all $x, y \in \mathrm{dom} f$, and θ with $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y).$$

A convex optimization problem is one of the form

$$\begin{aligned} \min \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i, \quad i = 1, ..., m. \end{aligned}$$

Here $x \in \mathbb{R}^n$ and both the objective function $f_0(x) : \mathbb{R}^n \mapsto \mathbb{R}$ and the constraint functions $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ are convex functions. A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is called *quasiconvex* if its domain and all its sublevel sets

$$\{x \in \mathrm{dom} f \mid f(x) \leq \alpha\}$$

for all $\alpha \in \mathbb{R}$ are convex. All convex functions are also quasiconvex, but the opposite is not necessarily true. Such an example is given in Figure 1. The following two properties of quasiconvex functions will be play an important in the proceeding sections.

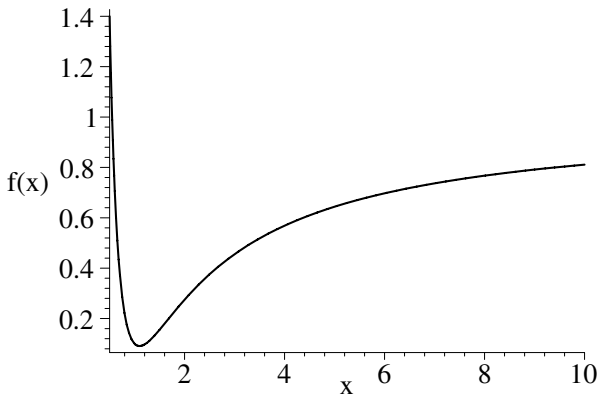


Figure 1: *A quasiconvex function f on $\mathbb{R}$, which is not convex. All sublevel sets of f are convex and in the one-dimensional case, these sets are intervals.*

Lemma 2.1 (Quasiconvex functions).

1. *If $f_1(x), ..., f_m(x)$ are quasiconvex functions, then $\max_i f_i(x)$ is also quasiconvex.*

2. *Let $f_i(x)$, $i = 1, 2, 3$ be affine functions, i.e., $f_i(x) = a_i^T x + b_i$. Then*

$$\frac{f_1(x)^2 + f_2(x)^2}{f_3(x)^2}$$

with domain $\{ x \mid f_3(x) > 0 \}$ is quasiconvex.

Proof. We need to prove that the sublevel sets are convex. 1. For any α, the sublevel set $\{ x \mid \max_i f_i(x) \leq \alpha \}$ is equal to $\{ x \mid f_i(x) \leq \alpha, i = 1, \ldots, m \} = \bigcap_i \{ x \mid f_i(x) \leq \alpha \}$. Thus the set can be written as the intersection of m convex sets and it is therefore convex.
2. Consider the sublevel set

$$\left\{ x \mid \frac{f_1(x)^2 + f_2(x)^2}{f_3(x)^2} \leq \alpha, f_3(x) > 0 \right\}$$

with $\alpha \geq 0$ (otherwise the set is empty) or equivalently $\{ x \mid f_1(x)^2 + f_2(x)^2 \leq f_3(x)^2 \alpha, f_3(x) > 0 \}$. Since $f_3(x)$ is positive, the set can be expressed using the standard Euclidean norm $|| \cdot ||$ as

$$\{ x \mid ||[f_1(x), f_2(x)]|| \leq \sqrt{\alpha} f_3(x), f_3(x) > 0 \}$$

which is a convex (positive) cone. □

Given that the objective function $f_0(x)$ is quasiconvex and that the constraint functions are convex, the problem may have locally optimal points, but still the global optimum can be efficiently computed via a sequence of convex feasibility problems. Let f_0^* denote the (unknown) optimal value of the quasiconvex object function. Given $\gamma \in \mathbb{R}$, if the convex

feasibility problem[1]

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & f_0(x) \leq \gamma \\ & f_i(x) \leq b_i, \quad i = 1, ..., m, \end{array} \tag{1}$$

is feasible, then we have $f_0^* \leq \gamma$. Conversely, if the above problem is infeasible, then we can conclude $f_0^* > \gamma$. Thus we can check whether the optimal value f_0^* is less or more than a given value γ. This observation is the basis of a simple algorithm for solving quasiconvex optimization problems using *bisection*, solving a convex feasibility problem at each step. It works in the following way.

Algorithm 2.1 (Bisection).

given: *optimal value* $f_0^* \in [\gamma_l, \gamma_u]$ *and tolerance* $\epsilon > 0$.
repeat
1. $\gamma := (\gamma_l + \gamma_u)/2$.
2. *Solve the convex feasibility problem (1).*
3. **if** *feasible,* $\gamma_u := \gamma$, **else** $\gamma_l := \gamma$.

until $\gamma_u - \gamma_l \leq \epsilon$.

A particular class of convex optimization problems, where the constraint functions are of the form

$$||A_i x + b_i|| \leq c_i^T x + d_i$$

are called *second order cone programs* (SOCP). Here $A_i \in \mathbb{R}^{n_i \times n}$, $b_i \in \mathbb{R}^{n_i}$, $c_i \in \mathbb{R}^n$ and $d_i \in \mathbb{R}$. The unit second-order, convex cone of dimension k is defined as

$$\mathcal{C}_k = \left\{ \begin{bmatrix} u \\ t \end{bmatrix} \mid u \in \mathbb{R}^{k-1}, t \in \mathbb{R}, ||u|| \leq t \right\}.$$

The reason for the name is that the set of points satisfying a second-order cone constraint is the inverse image of the unit second-order cone under an affine mapping:

$$||A_i x + b_i|| \leq c_i^T x + d_i \Leftrightarrow \begin{bmatrix} A_i \\ c_i^T \end{bmatrix} x + \begin{bmatrix} b_i \\ d_i \end{bmatrix} \in \mathcal{C}_{n_i}$$

and hence the SOCP is a convex optimization problem.

SOCP includes linear programming (LP) as a special case. On the other hand, it is less general than semidefinite programming (SDP). Solving SOCPs via SDP is not a good idea, however. The time complexity is much better for an SOCP algorithm than for an SDP algorithm [8].

3. The Triangulation Problem

We will start with one of the simplest geometric reconstruction problems where the goal is to infer the 3D structure given measured image points. Still, it is a fundamental problem in computer vision and there has been so far no satisfactory solution published for more than two views.

[1]The feasibility problem has no objective function, only convex constraints.

3.1. Problem Formulation

Let P_i, $i = 1, ..., m$, denote a set of 3×4 camera matrices and u_i the measured image points, all represented in homogeneous coordinates. Further, let $U = [x, 1]^T = [x_1, x_2, x_3, 1]^T$ denote the unknown 3D point. This leads us to the following minimization problem:

$$\begin{array}{ll} \min & \max_i d(u_i, P_i U(x)) \\ \text{subject to} & \lambda_i(x) > 0 \qquad i = 1, ..., m. \end{array} \tag{2}$$

Here $d(\cdot, \cdot)$ is the Euclidean image distance and $\lambda_i(x)$ is the depth of the point in image i. The inequality constraint makes sure that the point is in front of the camera. Given a perspective camera model, it follows easily that the squared image distance is a rational function of x:

$$d(u, PU(x))^2 = \frac{f_1(x)^2 + f_2(x)^2}{\lambda(x)^2},$$

where $f_1(x)$, $f_2(x)$ and $\lambda(x)$ are affine functions in x and with coefficients determined by u and P.

From Lemma 2.1, we know that $d(u, PU(x))^2$ is quasiconvex and trivially, so is $d(u, PU(x))$. Using the max-property in Lemma 2.1, it follows that problem (2) is indeed a quasiconvex optimization problem.

Remark. In the case of uncalibrated cameras, it may not be possible to determine the region of space that lies in front of all cameras. As pointed out in [3], the m principal planes - that is the plane of each camera consisting of points that map to infinity in the image - divide projective space $\mathbb{P}^3$ into $M_m = \binom{m}{3} + \binom{m}{1}$ regions. To find the minimum of the cost-function, it is necessary to find the minimum of each of the M_m regions. Once it is known that some point U lies in one of these regions, all other points must also lie in this region.

3.2. An Improved Bisection Method

Suppose that γ is an upper bound of the objective function in problem (2). It follows immediately that this upper bound also holds for each residual, $d(u_i, P_i U(x)) \leq \gamma$ for $i = 1, ..., m$. Revisiting the proof of Lemma 2.1, we see that this inequality can be formulated as a second order cone constraint using the convex cone $\mathcal{C}_3$. Based on these observations, problem (2) can be reformulated as the following problem:

$$\begin{array}{lll} \min & \gamma & \\ \text{subject to} & ||[f_{1i}(x), f_{2i}(x)]|| \leq \gamma \lambda_i(x) & \\ & \lambda_i(x) > 0 & i = 1, ..., m. \end{array} \tag{3}$$

If γ is considered to be known, the above problem can also be regarded as an SOCP feasibility problem, cf. (1):

$$\begin{array}{lll} \text{find} & x & \\ \text{subject to} & ||[f_{1i}(x), f_{2i}(x)]|| \leq \gamma \lambda_i(x) & \\ & \lambda_i(x) > 0 & i = 1, ..., m. \end{array} \tag{4}$$

Assume that the optimal γ^* is lower than some threshold of γ_u pixels, then evidently $\gamma^* \in [0, \gamma_u]$. Now, algorithm 2.1 can be directly applied, by solving the convex feasibility problem (4) at each iteration. However, we will make a small adjustment which will accelerate the bisection scheme significantly. Every time a feasible solution x is obtained for a given γ, one can compute the actual maximum image distance $\max_i d(u_i, P_i U(x))$ and this bound is always less than or equal to γ. In practice, we have found that it is often close to γ^*. So, instead of *step 3* in the standard bisection algorithm 2.1, it is better to replace it with:

3'. **if** feasible, $\gamma_u := \max_i d(u_i, P_i U(x))$ **else** $\gamma_l := \gamma$.

4. Projective Transformations and Projections

Projective geometry is a cornerstone of modern vision geometry. The basic tools for describing perspective mappings are projective transformations and projections. In this section we will show how the L_∞-framework can be applied to estimate such mappings. Although applications for mappings of higher dimensions than three exist in the vision literature [18], we will concentrate on plane-to-plane mappings, i.e., $\mathbb{P}^2 \mapsto \mathbb{P}^2$ and projections $\mathbb{P}^3 \mapsto \mathbb{P}^2$.

4.1. Points on a Plane

Let U_i, $i = 1, \ldots, m$ denote a set of planar points in space, represented by homogeneous plane coordinates. Given corresponding image features u_i, $i = 1, \ldots, m$, also represented by homogeneous coordinates, the two point sets are related by the relation $u_i \simeq H U_i$ where H is a projective transformation (also called a homography) represented by a 3×3 matrix. Let $H = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & 1 \end{bmatrix}$ and suppose the point coordinates are oriented in such a way as to comply with the positive depth constraint. Analogous to (2), the problem at hand becomes:

$$\begin{array}{ll} \min & \max_i d(u_i, H(x)U_i) \\ \text{subject to} & \lambda_i(x) > 0 \qquad i = 1, ..., m. \end{array} \quad (5)$$

The only difference compared to the triangulation problem is that $x \in \mathbb{R}^8$ and that the coefficients of the affine functions $f_{1i}(x)$, $f_{2i}(x)$ and $\lambda_i(x)$ in problem (3) are now determined by U_i and u_i. The global solution can be obtained with the bisection scheme, solving the SOCP feasibility problem (4) at each iteration.

Given image correspondences in two views of a set of (at least) four coplanar 3D points, the above procedure can also be used to estimate the inter-image homography (though the method is not symmetrical since all errors are assumed to be in one image).

4.2. Camera Resectioning

Another important problem is that of solving for camera pose given known 3D points and measured image points, which is also known as camera resectioning.

Let U_i denote a set of 3D points, and u_i the corresponding image points for $i = 1, \ldots, m$, and as usual represented by homogeneous coordinates. The objective is to find a 3×4 projection matrix P such that $u_i \simeq P U_i$. Similar to the homography above, the projection matrix can be parametrized by $P = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & 1 \end{bmatrix}$. Now, $x \in \mathbb{R}^{11}$ and the corresponding affine functions $f_{1i}(x)$, $f_{2i}(x)$ and $\lambda_i(x)$ in problem (3) are determined by U_i and u_i.

4.3. A Further Improved Bisection Method

The bisection method as described in algorithm 2.1 including the improvement in Section 3.2 can be further accelerated in terms of the number of iterations.

Note that in the homography and camera resectioning problems, the depth functions $\lambda_i(x)$, $i = 1, \ldots, m$ depend only on the last row of the homography matrix H and the projection matrix P, respectively. So, given a feasible solution x, one can do better than setting $\gamma_u := \max_i d(u_i, P_i U(x))$. If the variables of $\lambda_i(x)$ are considered to be fixed (or known), one can estimate the remaining variables in x *and* γ, simultaneously, by solving problem (3) which now has become a standard SOCP problem.

5. Multiview Geometry

We now turn to reconstruction problems with an arbitrary number of points and cameras.

5.1. Cameras with Known Rotation

In order to be able to apply the L_∞-framework for multiview reconstruction, we first assume that the rotational part of each camera is determined in advance. There are several scenarios where this is a reasonable assumption. For example, where the cameras are known to be purely translating or the rotation angles can be obtained from another sensor. Another setting is where the rotation matrix is pre-computed from an independent method, cf. [17].

Let $P = [\, R \;\; t \,]$ where R is a 3×3 matrix, assumed to be known, and $t = [\, x_1, x_2, x_3 \,]^T$ an unknown 3-vector. Further, let $U = [\, x_4, x_5, x_6, 1 \,]^T$ represent an unknown 3D point and u the measured image point. Then, the squared image residual can be expressed as $\frac{f_1(x)^2 + f_2(x)^2}{\lambda(x)^2}$ where $f_1(x)$, $f_2(x)$ and $\lambda(x)$ are affine functions with coefficients determined by R and u. Hence, the problem of recovering multiple instances of (i) camera translations and (ii) 3D points is a quasiconvex problem and can thus be solved via a series of SOCP feasibility proplems using the bisection method of algorithm 2.1.

In the above parametrization, there is a coordinate ambiguity. The reconstructed parameters are determined up to an unknown translation and scaling. Experimentally, it has been observed that the SOCP optimization may transform the coordinates to extreme values, thereby losing numerical accuracy. One possible way to fix the coordinate system is to set the coordinates of the first 3D point to $U_1 = [\,0, 0, 0, 1\,]^T$ and scale the first camera translation vector to $t_1 = [\,*, *, 1\,]^T$ (where $*$ denotes an unknown variable). This ensures that the positive depth constraint is fulfilled as well.

Remark. In constrast to many other methods, it is not necessary that all points are visible in all images.

5.2. Using a Reference Plane

By assuming that a reference plane is visible in all images, it is possible to estimate camera positions and 3D points up to an unknown projective transformation in closed form. The idea was pioneered in [11] where a linear method was developed based on an algebraic cost-function. We show here that it is possible to get globally optimal solutions based on the L_∞-norm. The exposition is by necessity brief.

Suppose that (at least) four points on a reference plane in 3D space are visible in all views. Then, it is possible to compute inter-image homographies between any two views with, for example, the method described in Section 4.1. Denote the inter-image homography mapping points from image 1 to image i by H_i for $i = 2, \ldots, m$ and define $H_1 = I$. Without loss of generality, one can choose a projective coordinate system such that the reference plane is given by the plane at infinity, denoted by Π_∞[2]. Then, it follows that the camera matrices are given by

$$P_i = \left[\begin{array}{cc} H_i & t_i \end{array}\right], \quad i = 1, \ldots, m,$$

where t_i is an unknown 3-vector. By parametrizing an unknown 3D point by $U = [\,x_1, x_2, x_3, 1\,]^T$ which is *not* on the reference plane, we have a problem with exactly the same appearance as the one described in Section 5.1. Hence, reconstructing cameras and 3D points given a reference plane is also a quasiconvex problem.

Note that 3D points on the reference plane need to be parametrized by $U = [\,x_1, x_2, x_3, 0\,]^T$. Therefore, it is required that all points are classified according to whether they belong to the reference plane or not. One way to determine this classification is to use the inter-image homographies. Points close the reference plane will have large magnitudes in contrast to points far away from it. Unlike the algebraic method of [11], our SOCP method is much less sensitive to this coordinate scaling, since the L_∞-optimization criterion is invariant to the world coordinate system. In practice, we have not encountered any problems due to this phenomenon in the SOCP optimization, even though coordinates may be relatively large.

[2] A point in $\mathbb{P}^3$ lies on Π_∞ if and only if the last coordinate in the homogeneous coordinate vector is zero.

Figure 2: *Examples of images in the corridor (top row) and the dinosaur (bottom row) sequences.*

In an actual image sequence, the reference plane can be either finite or infinite (which should not be confused with the Π_∞-parametrization). Typically, four coplanar points determine a finite reference plane. Three orthogonal vanishing points can be used to determine a reference plane at infinity. See [11] for further details.

6. Experimental Validation

In order to test the proposed framework, we have made extensive use of two publicly available sequences with given feature correspondences[3]. The first sequence consists of 11 images in a corridor, see Figure 2. There are 104 point correspondences visible in all images. The other image set is a turntable sequence of a dinosaur, containing 36 images and in total 328 image correspondences with lots of occlusions. In the experiments, the number of views and points have artificially been varied to test the performance in different settings.

For comparison, we also applied standard linear algorithms and bundle adjustment [5] (which optimizes the L_2-norm) to exactly the same data. Proper normalization has also been done as a preprocessing step for the linear algorithms. It is not evident by which norm the algorithms should be compared. On one hand, we wish to show that the bisection scheme computes the optimal estimates with respect to the L_∞-norm. On the other hand, the L_2-norm has a statistical meaning and it would be valuable if our L_∞-estimates perform well with respect to this norm as well. Therefore, for the first experiment, we give results for both measures. The Root Mean Squares (RMS) errors of the reprojected and measured points are reported. The bundle adjustment has been initialized with both the linear algorithm and the bisection method, and the one with lowest RMS error is kept.

Implementation. All the routines for L_∞-optimization have been collected in a toolbox which is publicly available[4].

[3] See http://www.robots.ox.ac.uk/∼vgg/data.html.

[4] See http://www.maths.lth.se/matematiklth/personal/fredrik/download.html.

The bisection algorithm 2.1 (including the modification in Section 3.2) for the proposed applications has been implemented under the Matlab environment using SeDuMi [12] which is a toolbox for optimizing over convex cones.

Typically, the interval length of $[\gamma_l, \gamma_u]$ is less than 10^{-5} pixels within 5-10 iterations of the bisection method. Due to this rapid convergence, the modification described in Section 4.3 has not been evaluated. The computation times, that is, the cputime for one call to SeDuMi, on a Pentium 4 with 2.8 GHz for the SOCP feasibility problem (4) vary from 0.05 s for three-view triangulation to 1 s for a multiview reference plane problem with 36 cameras and 2270 $\mathcal{C}_3$-cone constraints (one cone for each visible image point).

6.1. Triangulation

In order to test the triangulation method, the camera matrices need to be pre-computed and they have been obtained with the reference plane technique (including bundle adjustment), cf. Section 6.4.

The results for triangulation are given in Figure 3. As expected, when the average L_∞-reprojection error is used to compare the three algorithms, the L_∞-method is best. This is in accordance with the theory as the estimates should be globally optimal. When compared using RMS errors, the bundle adjustment method optimizing the L_2-method performs best, also as expected.

6.2. Points on a Plane

In Figure 4, the errors for inter-image homography estimation are shown with respect to the first image. In the corridor sequence, 23 points out of the 104 points can be found on the left frontal wall, and they were used as input data. In the dinosaur images, the coplanar points are located on the turntable and the number of visible points vary between 6 and 12 throughout the sequence. All three methods are very similar in performance.

6.3. Camera Resectioning

In order to test the camera pose estimation, the 3D structure obtained from the reference plane technique was used, cf. Section 6.4. Again, the L_∞-estimates are comparable to those of L_2-minimization and the linear method is remarkably good in this case, cf. Figure 5.

6.4. Multiview Geometry

Given the computed inter-image homographies based on the L_∞-norm (Section 6.2), it is now possible to estimate the camera translations and the remaining 3D points for the two sequences (Section 5.2). This works amazingly well - the RMS errors are low and the global estimates are obtained within seconds. Recall, for instance, that the dinosaur sequence contains 36 images and hundreds of points.

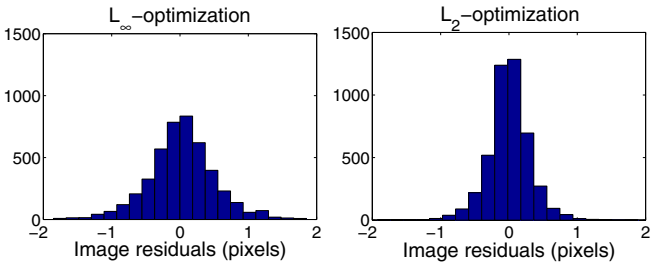


Figure 6: *Histograms of image residuals for the dinosaur sequence with* 36 *images.*

The RMS errors for the corridor and dinosaur sequences are 0.43 pixels and 0.49 pixels, respectively. The corresponding L_∞-errors are 1.91 pixels and 1.86 pixels. After applying bundle adjustment over *all* structure and motion parameters, the corresponding RMS errors become 0.25 pixels and 0.31 pixels, and L_∞-errors become 1.46 pixels and 2.16 pixels, respectively. Only *four* bundle adjustment steps were needed to obtain the minimum for both sequences. In Figure 6, histograms of the image residuals of x- and y-coordinates in the dinosaur sequence are plotted. Notice that for the L_2-optimization the residuals become more peaked compared to the L_∞-optimization.

7. Discussion

A geometric framework for computing globally optimal estimates with respect to the L_∞-norm has been introduced. The estimates are invariant with respect to projective transformations of the world coordinate system as well as similarity transformations in the image plane. Unlike algebraic methods, the cost-function has a clear geometric meaning and there is no need for data normalization. Iterative refinement techniques, such as bundle adjustment, can be used to improve the estimates with respect to the L_2-norm. Though, in practice, we have found that the L_∞-estimates are already quite good.

An obvious criticism of using the L_∞-norm is its sensitivity to outliers, as pointed out by [3]. In a sense, we are fitting the noisiest data. However, it is undeniable that outliers are also fatal to the ordinary L_2-norm as well as linear algorithms. An interesting research direction would be to use L_∞-optimizing for detecting outliers. The SOCP feasibility problem is well-suited for that purpose. Another criticism is that in certain situations, a linear algorithm may yield better estimates with respect to the L_2-norm.

An advantage of the L_∞-approach is that we have a fair assurance of how well the data fits the model, while with L_2-norm the data may be good, but the optimization has fallen into a local minimum. Similarly with linear algorithms, the data may be good, but the algebraic cost-function might produce an unreasonable estimate.

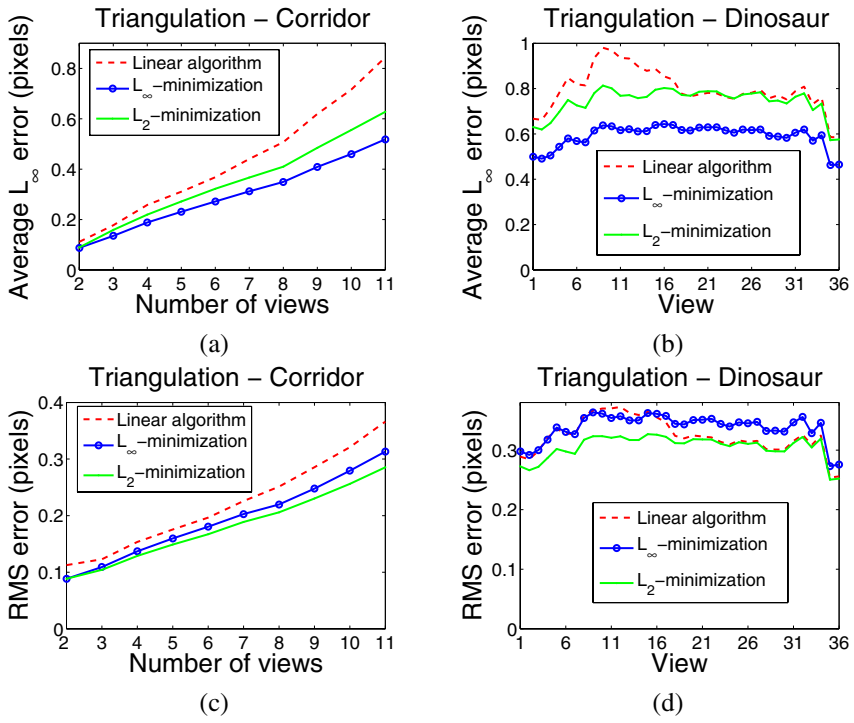


Figure 3: *Triangulation results for the corridor and dinosaur sequences. In (a) and (b), the L_∞-errors are graphed and in (c) and (d), the RMS errors. The graphs in (a) and (c) show the result with varying number of camera views. The graphs in (b) and (d) show the errors for all visible points* per *view.*

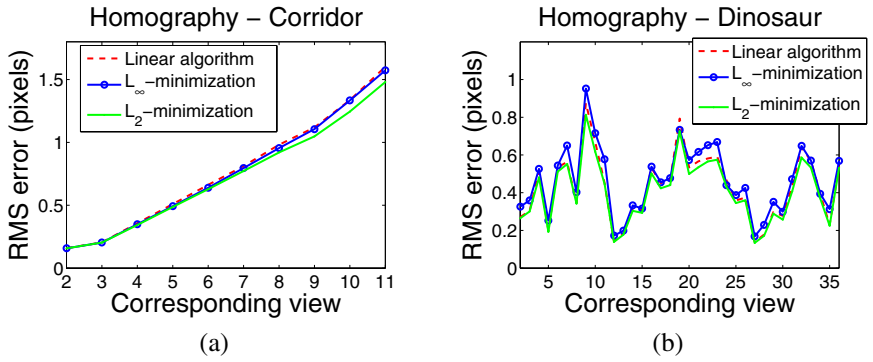
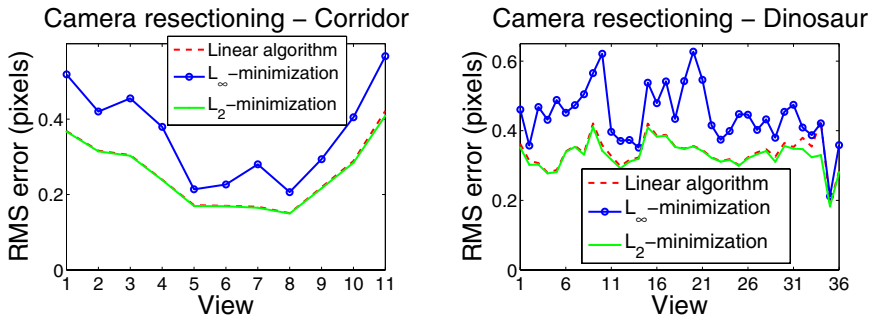


Figure 4: *Graphs for inter-image homography estimation with respect to the first image of the sequence. The reason why the errors increase in (a) is a bit unclear. It may be due to drift in the feature extraction when the points were tracked or non-linear effects not modelled by the pinhole camera model.*



(a) (b)

Figure 5: *Estimation of camera poses. In both the corridor and the dinosaur sequence, all three methods perform well. The linear method is in this case very close to the results of bundle adjustment.*

Acknowledgements

This work has benefited from many discussions with Manmohan Chandraker, Sameer Agarwal, David Kriegman and Didier Henrion. Financial support was provided by the U.C. Micro Program, the Swedish Research Council and the SSF-funded Viscos.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] 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.

[3] R. Hartley and F. Schaffalitzky. L-∞ minimization in geometric reconstruction problems. In *Conf. on Computer Vision and Pattern Recognition*, volume I, pages 504–509, Washington DC, USA, 2004.

[4] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

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

[6] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Scandinavian Conf. on Image Analysis*, pages 963–968, Lappeenranta, Finland, 1997.

[7] F. Kahl. Globally Optimal Estimates for Geometric Reconstruction Problems. In *Int. Conf. on Computer Vision*, Beijing, China, 2005.

[8] M.S. Lobo, L. Vandenberghe, S.P. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Its Applications*, 284:193–228, 1998.

[9] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[10] D. Nistér. *Automatic dense reconstruction from uncalibrated video sequences*. PhD thesis, Royal Institute of Technology KTH, Sweden, 2001.

[11] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery using a reference plane. *Int. Journal of Computer Vision*, 49(2/3):117–141, 2002.

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

[13] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. on Computer Vision*, pages 709–720, Cambridge, UK, 1996.

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

[15] P.H.S. Torr and A.W. Fitzgibbon. Invariant fitting of two view geometry. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5):648–650, 2004.

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

[17] M. Uyttendaele, A. Criminisi, S.B. Kang, S. Winder, R. Szeliski, and R. Hartley. Image-based interactive exploration of real-world environments. *Computer Graphics and Applications*, 24(3):52–63, 2004.

[18] L. Wolf and A. Shashua. On projection matrices $P^k \mapsto P^2$, $k = 3, \ldots, 6$, and their applications in computer vision. In *Int. Conf. on Computer Vision*, volume I, pages 412–419, Vancouver, Canada, 2001.