

Robust Fitting for Multiple View Geometry

Olof Enqvist, Erik Ask, Fredrik Kahl, and Kalle Åström

Centre for Mathematical Sciences, Lund University
<http://www.maths.lth.se/vision/>

Abstract. How hard are geometric vision problems with outliers? We show that for most fitting problems, a solution that minimizes the number of outliers can be found with an algorithm that has polynomial time-complexity in the number of points (independent of the rate of outliers). Further, and perhaps more interestingly, other cost functions such as the truncated L_2 -norm can also be handled within the same framework with the same time complexity.

We apply our framework to triangulation, relative pose problems and stitching, and give several other examples that fulfill the required conditions. Based on efficient polynomial equation solvers, it is experimentally demonstrated that these problems can be solved reliably, in particular for low-dimensional models. Comparisons to standard random sampling solvers are also given.

1 Introduction

The ability to handle outliers in multiple view geometry problems is a major challenge. It is well-known that least-squares methods will inevitably break down due to erroneous correspondences, so there is a need for methods that are robust to noisy image measurements. A common choice is to try to maximize the number of inliers, that is, points within some error tolerance. More ideally, one would like to use a robust cost function such as the truncated L_2 -norm. In this paper, we analyze the problem of robust fitting and show that many geometric problems are in fact tractable under various cost functions. The validity of our approach is experimentally demonstrated on three application problems; see Figure 1.

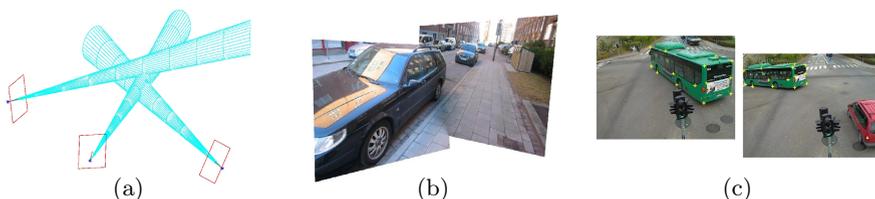


Fig. 1. (a) *Triangulation.* An image point gives rise to a cone $\{\theta \in D : r_i(\theta) \leq \epsilon\}$. (b) *Stitching.* Images taken by a stationary camera can be stitched together. (c) *Planar motion.* A surveillance camera views an object moving in a plane.

1.1 Related Work

One of the most successful approaches to robust fitting is RANSAC, originally proposed in [1] and later refined in various aspects, e.g., [2]. In RANSAC a series of minimal subproblems are solved and the solutions are evaluated on the full problem. Normally, noise-free solutions to the subproblems are sought but there are also cases where the subproblem is optimized in some sense, e.g. [3]. It is implicitly assumed that a good solution can be found as a solution to one of the subproblems. In this paper it is shown that by using a different type of subproblems, an optimal solution can often be guaranteed.

A different paradigm that has also turned out to be useful in this context is the L_∞ -framework [4]. In [5], a heuristic relaxation method is proposed to remove outliers. In [6], it is shown how to detect outliers but the method tends to remove a lot of inliers as well. Further extensions in this direction have been explored in [7, 8]. The approach works well for large-scale problems with few outliers, but cannot handle large rates of outliers.

The most similar works to ours include [9–13] where the aim is to develop algorithms which provably maximizes the number of inliers. In [9, 10], branch-and-bound techniques are developed which have exponential worst-time complexity. In [11], registration problems dealing mainly with 2D transformations are considered. In the case of linear constraints on the transformation space (e.g., when optimizing over translations), it is concluded that in order to obtain the optimal solution, one only needs to examine the intersections of the constraints. This is equivalent to what our approach boils down for this specific setting. However, in the case of non-linear constraints (more specifically, in the case of rotations), specialized solutions are proposed which do not apply to the general setting that we consider. In [12], a triangulation method is presented, but it is only practical for a few outliers due to its high computational complexity. For quasiconvex residual functions, an $O(n^{d+2})$ algorithm is given in [13], where n is the number of points and d the dimension of the model. We improve on this result by showing it is possible to solve the same problem in $O(n^{d+1})$. Further, our result holds for a larger set of residual functions and we can model other cost functions than the cardinality of the inlier set.

2 Problem Formulation

Let $\{r_i(\theta)\}_{i=1}^n$ be a set of non-negative residual functions. Consider the following optimization problem,

$$\min_{\theta} \sum_{i=1}^n \ell(r_i(\theta)), \quad (1)$$

where $\theta \in D \subseteq \mathbb{R}^m$. Here $\ell(x)$ is a loss function, for example, $\ell(x) = 1$ if $x > \epsilon$ and 0 otherwise; see Figure 2(a). Hence, with this loss function, (1) is simply minimizing the number of outliers. We will require that the loss function ℓ is non-decreasing and piecewise constant with a finite number of discontinuity points. This may seem like a severe restriction, but we can in fact approximate

any truncated norm-function to arbitrary precision. An example is the truncated L_2 -norm given in Figure 2(b).



Fig. 2. (a) The number of inliers (i.e., residual errors below ϵ) is minimized when the loss function is a step function. (b) The truncated L_2 -norm (blue) can be approximated with a non-decreasing piecewise constant function (red).

Before stating our main theorem and developing an algorithm for solving the above problem, we will give a few example applications.

Example 1 (Line fitting). Given a set of points (x_i, y_i) , $i = 1, \dots, n$, we wish to fit a line to the data. In this case the parameter space is a two-dimensional manifold or more precisely the cylinder $D = \{\theta \in \mathbb{R}^3 : \theta_1^2 + \theta_2^2 = 1\}$. The residual functions are computed as

$$r_i(\theta) = |x_i\theta_1 + y_i\theta_2 + \theta_3|.$$

Example 2 (Triangulation). Given a set of 3×4 camera matrices P_i and corresponding image points (x_i, y_i) , $i = 1, \dots, n$, the goal is to reconstruct a 3D point. The residual functions measuring reprojection errors can be written

$$r_i(\theta) = \left\| \left(x_i - \frac{a_i^T \theta + \bar{a}_i}{c_i^T \theta + \bar{c}_i}, y_i - \frac{b_i^T \theta + \bar{b}_i}{c_i^T \theta + \bar{c}_i} \right) \right\|, \tag{2}$$

where (a_i^T, \bar{a}_i) , (b_i^T, \bar{b}_i) and (c_i^T, \bar{c}_i) denote the first, second and third row, respectively, of P_i . The domain is the set of points in front of all cameras

$$D = \{\theta \in \mathbb{R}^3 : c_i^T \theta + \bar{c}_i \geq 0, i = 1, \dots, n\}.$$

All points that satisfy $r_i(\theta) \leq \epsilon$ form a convex cone; see Figure 1(a).

Example 3 (Rigid registration). Let (x_i, y_i) and (x'_i, y'_i) be corresponding 2D points, $i = 1, \dots, n$, and suppose we wish to estimate a rigid transformation such that the first set of points is mapped to the other. The residual functions are given by

$$r_i(\theta) = \left\| \begin{pmatrix} \theta_1 & -\theta_2 \\ \theta_2 & \theta_1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} \theta_3 \\ \theta_4 \end{pmatrix} - \begin{pmatrix} x'_i \\ y'_i \end{pmatrix} \right\|.$$

The parameter space is (as in Example 1) a cylinder, but it is now a three-dimensional manifold $D = \{\theta \in \mathbb{R}^4 : \theta_1^2 + \theta_2^2 = 1\}$.

3 Main Result

In this section we will examine the set of minimizers to (1). Specifically, we will define a non-empty subset of the minimizers and show that these points can be found by analyzing subproblems of at most d residual functions, where d is the dimension of the parameter space. The number of such subproblems is $\binom{n}{d}$. For low values of d , an exhaustive search over these sets is tractable.

Let the discontinuity points of loss function ℓ be denoted by $\epsilon_1, \dots, \epsilon_k$; cf. Figure 2. As a consequence, the sum in (1) can only attain a finite number of different values and hence there exists at least one minimizer for (1). Let θ^* be such a minimizer and consider the set

$$\Omega^* = \bigcap_{i=1}^n \{\theta : \ell(r_i(\theta)) \leq \ell(r_i(\theta^*))\}. \tag{3}$$

Clearly this set is non-empty since it contains θ^* . Moreover, any θ in this set is a minimizer to (1). Our aim is to devise an algorithm that is guaranteed to find at least one such θ . We start by simplifying (3).

First we note that for each i such that $r_i(\theta^*) > \epsilon_k$, that is, the value is larger than any discontinuity point, the constraint $\ell(r_i(\theta)) \leq \ell(r_i(\theta^*))$ does not restrict θ and can be dropped. Let I^* denote the indices such that $r_i(\theta^*) \leq \epsilon_k$. With the 0 – 1 loss function, this is simply the set of inliers. Moreover, the fact that ℓ is piecewise constant makes it possible to simplify the description, since

$$\ell(r_i(\theta)) \leq \ell(r_i(\theta^*)) \Leftrightarrow r_i(\theta) \leq \epsilon_{j_i}$$

where ϵ_{j_i} takes one of the values $\epsilon_1, \dots, \epsilon_k$. This allows us to rewrite Ω^* as

$$\Omega^* = \bigcap_{i \in I^*} \{\theta : r_i(\theta) \leq \epsilon_{j_i}\}.$$

Now consider the optimization problem,

$$\min_{\theta} f(\theta) \quad \text{s.t.} \quad r_i(\theta) \leq \epsilon_{j_i} \quad \text{for } i \in I^*. \tag{4}$$

where f is an auxiliary goal function. We note that a solution to this problem will also solve the original problem.

3.1 Main Theorem

To proceed we need some regularity conditions for problem (4).

Condition 1. *The goal function f is continuously differentiable and there exists a minimizer to (4). Moreover, the residual functions are continuously differentiable at the minimizer and satisfy a constraint qualification for the Karush-Kuhn-Tucker (KKT) constraints [14].*

A constraint qualification is a standard condition in the optimization literature and we will discuss it in more detail later. The importance of constraint qualifications is to guarantee that, by examining only KKT points, we do not lose out on local minima and, hence possibly, global optimal solutions.

Theorem 1. *Assume that the domain D satisfies $D = \mathbb{R}^d$ and Condition 1 is fulfilled. Then a minimizer of (4) is also a KKT point for a subproblem of size at most d , that is,*

$$\min_{\theta} f(\theta) \quad \text{s.t.} \quad r_i(\theta) \leq \epsilon_{j_i} \quad \text{for } i \in I, \tag{5}$$

where I of is a subset of $\{1, \dots, n\}$ of size $|I| \leq d$.

The following extension of the theorem of Caratheodory will be useful.

Lemma 1. *Let v and $e_i, i = 1, \dots, n$ be in \mathbb{R}^d and assume that*

$$v + \sum_{i=1}^n \mu_i e_i = 0 \quad \text{with } \mu_i \geq 0. \tag{6}$$

Then there exists a subset I of $\{1, \dots, n\}$ of size $|I| \leq d$ such that

$$v + \sum_{i \in I} \bar{\mu}_i e_i = 0 \quad \text{with } \bar{\mu}_i > 0.$$

Proof. By (6), $-v$ is a positive linear combination of the e_i 's. By the theorem of Caratheodory, it can also be written as a positive linear combination of $d + 1$ of the e_i 's. After renumbering we can write this as

$$v + \sum_{i=1}^{d+1} \mu'_i e_i = 0.$$

Hence 0 is a positive linear combination of $d + 2$ vectors. Again by Caratheodory, it is also a positive linear combination of $d + 1$ of these vectors. If v is one of them we are done. Otherwise $\sum_{i=1}^{d+1} \mu''_i e_i = 0$. Now let $k = \min \mu'_i / \mu''_i$ and assume that the minimizer is $i = d + 1$,

$$0 = v + \sum_{i=1}^{d+1} \mu'_i e_i - k \sum_{i=1}^{d+1} \mu''_i e_i = v + \sum_{i=1}^d (\mu'_i - k\mu''_i) e_i.$$

Since $\bar{\mu}_i = \mu'_i - k\mu''_i$ is always non-negative, this completes the proof. □

Proof (Theorem 1). The KKT conditions state that in optimum

$$\nabla f + \sum_{i=1}^n \mu_i \nabla r_i = 0 \quad \text{with } \mu_i \geq 0$$

and that $\mu_i(r_i - \epsilon_{j_i}) = 0, i = 1, \dots, n$. By Lemma 1 there is a subset I of $\{1, \dots, n\}$ of size $|I| \leq d$ such that

$$\nabla f + \sum_{i \in I} \bar{\mu}_i \nabla r_i = 0 \quad \text{with } \bar{\mu}_i > 0.$$

Clearly, $\bar{\mu}_i(r_i - \epsilon_{j_i}) = 0$ holds so the optimum is a KKT point for the subproblem generated by I as well. □

3.2 An Algorithm

Theorem 1 implies that a solution to (4) – and hence a solution to our main problem (1) – is a KKT point for a subproblem of size (at most) d . To find the KKT points of this problem we need to work out which residual functions are involved and which thresholds ϵ_{j_i} are relevant. We will handle this by performing an exhaustive search over subsets and thresholds. Recall that there are n different residual functions and k different thresholds. Hence, there are $\binom{n}{d}$ different choices of residual functions and for each choice, the d thresholds can be chosen in k^d different ways. In total, there are $\mathcal{O}(k^d n^d)$ different cases to consider. For each of these we compute all n residuals. Hence the algorithm has complexity $\mathcal{O}(k^d n^{d+1})$, which is polynomial in the number of residuals, but exponential in the dimension of the parameter space. Note, however, that for a given problem type the dimension d is fixed. Algorithm 1 gives an overview.

Algorithm 1. Polynomial-Time Optimum Search

Given a set of residual functions, solve main problem (1)

For each subset I of $\{1, \dots, n\}$ of size $|I| \leq d$

For each choice of thresholds $\epsilon_{j_1}, \dots, \epsilon_{j_d}$.

Compute all KKT points, denoted θ_{KKT} , in (5).

For each θ_{KKT} ,

Compute all residuals $r_i(\theta_{\text{KKT}})$ and count the outliers.

If this is the lowest number of outliers so far, store θ_{KKT} .

4 Finding the KKT Points

Algorithm 1 requires a method to find all KKT points for a given subproblem (5). If the residual functions are quasiconvex (for example, as in triangulation) and the auxiliary goal function is convex we can do this by using standard methods for convex optimization [4]. A faster approach can be obtained if the KKT points can be described using polynomial equations and this is often the case in multiple view geometry. A KKT point is characterized by (possibly after renumbering)

$$\begin{aligned} \nabla f(\theta) + \sum_{i=1}^d \mu_i \nabla r_i(\theta) &= 0, \\ \mu_i(r_i(\theta) - \epsilon_{j_i}) &= 0 \quad \text{for } i = 1, \dots, n. \end{aligned} \tag{7}$$

Constraints for which $\mu_i \neq 0$ are called active since (7) implies that equality holds for these constraints. A KKT point will have between 0 to d active constraints.

5 Application I: Triangulation

Given n cameras with known camera matrices we wish to determine the coordinates of a scene point $\theta \in \mathbb{R}^3$. The reprojection residual functions $r_i(\theta)$ are given by (2). To solve for the KKT points we need to enforce constraints of the type $r_i(\theta) \leq \epsilon_{j_i}$ which is a convex cone in \mathbb{R}^3 . Since the intersection of convex sets is convex, the optimization in (5) is over a convex set. Any convex set has a unique point with minimal norm, so we choose the auxiliary goal function to be $f = \|\theta\|^2$. This ensures that a minimizer exists. Moreover, the residual functions in (2) are continuously differentiable except in the camera centre. For the last part of Condition 1 we use Slater's condition [14]. It requires that the feasible set should contain an inner point. The feasible set in (5) is the intersection of a number of convex cones. One can prove that except in some degenerate configurations (of measure zero), this intersection has an inner point. It follows, that except in (unlikely) degenerate cases, the optimum of (5) is either a KKT point or a camera centre.

To find the KKT points we need to consider the active constraints $r_i(\theta) = \epsilon_{j_i}$, which can be rewritten as $g_i(\theta) = 0$, where

$$g_i(\theta) = (\theta^T (x_i c_i - a_i) + x_i \bar{c}_i - \bar{a}_i)^2 + (\theta^T (y_i c_i - b_i) + y_i \bar{c}_i - \bar{b}_i)^2 - \epsilon_{j_i}^2 (\theta^T c_i + \bar{c}_i)^2. \quad (8)$$

Clearly this is a quadratic equation in θ . As noted in Section 4 there will be up to $d = 3$ active constraints. If there are exactly three, then we get a system of three quadratic equations as in (8). As predicted by Bezout's theorem there will be up to 8 solutions. If there are only two active constraints, then we add an equation by forcing the gradients to be linearly dependent,

$$0 = \det (\nabla g_i \nabla g_j \nabla f) = \det (\nabla g_i \nabla g_j 2\theta),$$

which is a cubic equation. This also yields a reasonable system with 12 solutions. The case with one active constraint can be handled similarly. Finally we need to check $\theta = 0$, since then $\nabla f = 0$.

5.1 Experimental Validation

Algorithm 1 was implemented in MATLAB. The polynomial equations were solved using a generic polynomial solver based on [15]. Experiments were run on the well-known Dinosaur sequence using the step loss function, i.e. trying to minimize the number of outliers. We compare our results with a standard minimal solver. In this case we used the optimal two-view triangulation from [3]. To get a fair comparison that solver was tried for all possible choices of two views. Hence the

result is the best possible that could be achieved with RANSAC. Figure 3 shows a comparison to the proposed method. For most of the triangulated points in the Dinosaur sequence there are no outliers. We only consider cases with at least one outlier.

	Error threshold	
	1 pixel	2 pixels
Same result	108	51
1 outlier less	156	82
2 outliers less	10	2

Fig. 3. Comparison between the proposed method and the best possible result using a standard solver. For two different thresholds the number of times that a certain improvement was achieved are presented. For example, with a threshold of 1 pixel, the proposed method had one outlier less in 156 cases.

The running times for the generic solver were around 0.1 s per minimal case on a desktop computer. A specialized solver can solve the same problem in a few milliseconds [15].

6 Extension to Manifolds

A lot of interesting models are not natural to parameterize using the whole of \mathbb{R}^d . For example, rotations are most naturally parameterized using unit 4-vectors. Hence, we will study the case when the parameter space is a d -dimensional differentiable manifold.

Theorem 2. *Assume that the domain D is a d -dimensional differential manifold embedded in R^m ($d < m$) by $h_j(\theta) = 0, j = 1, \dots, m - d$, and that Condition 1 is fulfilled. Then a minimizer of (4) is also a KKT point for a subproblem of size at most d , that is,*

$$\begin{aligned} \min_{\theta} f(\theta) \quad \text{s.t.} \quad & r_i(\theta) \leq \epsilon_{j_i} \quad \text{for } i \in I \\ & h_j(\theta) = 0 \quad \text{for } j = 1, \dots, m - d, \end{aligned}$$

where I is a subset of $\{1, \dots, n\}$ of size $|I| \leq d$.

Proof. Consider a KKT point θ to the full problem. The KKT conditions state that in optimum

$$\nabla f + \sum_{i=1}^n \mu_i \nabla r_i + \sum_{j=1}^{m-d} \lambda_j \nabla h_j = 0, \tag{9}$$

where $\mu_i \geq 0$ and that $\mu_i(r_i - \epsilon_{j_i}) = 0, i = 1, \dots, n$. The fact that the parameter space is a d -dimensional manifold means that the gradients $\{\nabla h_j\}$ span a $(m - d)$ -dimensional subspace of R^m orthogonal to the d -dimensional tangent space at the KKT point. If we let P be an operator projecting to the tangent space, then

$$P\nabla f + \sum_{i=1}^n \mu_i P\nabla r_i = 0.$$

By Lemma 1, there is a set, I , with $|I| \leq d$ such that

$$P\nabla f + \sum_{i \in I} \bar{\mu}_i P\nabla r_i = 0, \tag{10}$$

and $\bar{\mu}_i(r_i(\theta) - \epsilon_{j_i}) = 0$ for all $i \in I$. Now consider $\nabla f + \sum_{i \in I} \bar{\mu}_i \nabla r_i$. Clearly it is orthogonal to the tangent space of the manifold since its projection (10) was zero. Hence it can be written as a linear combination of the ∇h_j 's,

$$\nabla f + \sum_{i \in I} \bar{\mu}_i \nabla r_i + \sum_{j=1}^{m-d} \bar{\lambda}_j \nabla h_j = 0.$$

We have proven that θ is a KKT point for (9). □

7 Application II: Stitching

Given two sets of unit 3-vectors we wish to estimate a rotation that maps one set onto the other. This problem typically arises when stitching images to panoramas. A rotation in \mathbb{R}^3 can be parametrized using 3 angles representing the rotation around each axis. This approach however involves trigonometric functions and is unsuitable in the framework of polynomial solving. A more tractable approach is to use the well-known quaternion representation of rotations in \mathbb{R}^3 . A rotation is represented by a unit 4-vector θ with $\|\theta\| = 1$. Similar to triangulation, Condition 1 will be fulfilled except for rare, degenerate instances.

To find the KKT points we need to consider the active constraints $r_i(\theta) = \epsilon_{j_i}$, which can be rewritten as $g_i(\theta) = 0$, where

$$g_i(\theta) = x_i R_\theta x'_i - \cos(\epsilon). \tag{11}$$

This is a quadratic equation in θ . As noted in Section 4 there will be up to $d = 3$ active constraints. If we have exactly three, we have three equations as in (11) and $\|\theta\|^2 = 1$ which is sufficient to solve for the unknowns. Since all the equations are quadratic in θ we get 16 solutions.

If only two constraints are active we know from (7) that two points satisfy (11) and by the embedding that $\|\theta\|^2 = 1$. This totals three equations. Just as in the triangulation case, we get the fourth equation by forcing the gradients to be linearly dependent,

$$0 = \det \left(\nabla f \quad \nabla(\|\theta\|^2 - 1) \quad \nabla g_i \quad \nabla g_j \right),$$

where i, j denotes the indexes of the active correspondences. Since we are free to choose the objective function $f(\theta)$ we set $f(\theta) = \theta_1$ giving $\nabla f = (1, 0, 0, 0)$ essentially reducing the size of the determinant. We get three quadratic equations and one cubic, so the number of solutions is 24. One can show that the case with one active constraint does not have to be handled.

7.1 Experimental Validation

A total of 95 image pairs were produced for various scenes. Feature correspondences were established using SIFT and the internal camera parameters were extracted from the EXIF tags. In each image pair, 10 point correspondences were selected randomly for the experiment. For the first experiment we used the step loss function in Figure 2(a), that aims to minimize the number of outliers. Figure 4(a) shows the improvement over the best solution achievable using RANSAC with a standard solver. In this case a standard solver is a two-point solver minimizing the angular error. For the second experiment, we use the approximative truncated L_2 norm; see Figure 2(b). The norm was truncated at 0.1° and approximated with three steps at 0.01° , 0.03° and 0.1° . Figure 4(b) shows a histogram over the improvement. The improvement is in terms of the sum of squared truncated errors, so the unit is degrees². Note that the exact loss function is used for the comparison and not the approximated version. The same standard solver was used for comparison but this time the solution was updated with respect to the truncated L_2 norm.

Like in the case of triangulation the generic polynomial solver required about 0.1 s per minimal case, but this could be reduced to a couple of milliseconds.

	Error threshold	
	0.1°	0.2°
Same result	45	70
1 outlier less	41	20
2 outliers less	7	4
3 outliers less	0	1

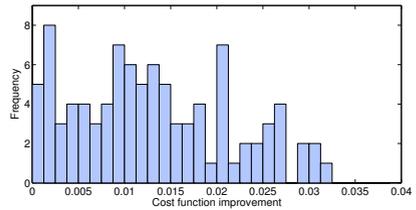


Fig. 4. (Stitching). (a) Improvement for the first loss function, i.e. minimizing the number of outliers. (b) Improvement for truncated L_2 norm.

8 Application III: Planar Motion

Consider a camera that is moving and rotating in a known plane – or equivalently a fixed camera observing an object that is moving and rotating in a known plane; cf. Figure 1(c). Without losing generality we can assume that the plane is $z = 0$. The camera matrix can be written,

$$P = S[R | t], \quad (12)$$

where S is a known rotation matrix, R is an unknown rotation about the z -axis and $t = (t_1, t_2, 0)$. The model has two degrees of freedom ($d = 2$) being a rotation angle and a translation direction in the plane. According to Theorem 2 we can find the optimum by considering subproblems with one or two residuals. It turns out that the relevant case is the one with two active residual constraints. Let $(x_i, y_i), (x'_i, y'_i)$, $i = 1, 2$ be the correspondences giving rise to those residuals. Quantities in the second camera will be primed.

All 3D points that project to points with distance ϵ to the point (x, y) can be written

$$(P\tilde{X})^T C P \tilde{X} = 0, \tag{13}$$

where \tilde{X} denotes homogeneous coordinates of the 3D points and

$$C = \begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ -x & -y & x^2 + y^2 - \epsilon^2 \end{pmatrix}.$$

There are two cases that we need to consider. If X is a point at infinity, then (13) is a constraint only on the camera rotation and the relevant rotations can be computed from a single point correspondence. Given this rotation another point correspondence is used to compute relevant translations. For the sake of brevity, we leave out the exact derivations for this case. The second case is the one with finite X . Using the form of P from (12), we can rewrite (13) as

$$(X + t)^T R^T S^T C S R (X + t) = 0.$$

Here X denotes standard coordinates in \mathbb{R}^3 . We note that $S^T C S$ is known and that this constrains $X + t$ to a cone. To eliminate t we consider the dual of this cone. A vector n is a possible normal to this cone if

$$n^T R^T S^T C^{-1} S R n = 0.$$

We define $N = S^T C^{-1} S$. We get a cone of this sort for the other view as well, denoted N' . For the reprojection error to be exactly ϵ and ϵ' in the two views, respectively, we need these cones to be tangent to each other and thus should contain a common normal n ; see Figure 5.

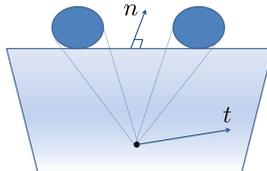


Fig. 5. For two cones to be tangent, they should have a common normal n . Note that the cones here have been translated to a common origin. The translation t should lie in the plane with normal n .

For two correspondences, we get

$$n_i^T R^T N_i R n_i = 0 \text{ and } n_i^T R'^T N'_i R' n_i = 0. \tag{14}$$

If we choose the original coordinate system wisely, we can set $R' = I$ and parameterize R with a unit 2-vector (a, b) adding the constraint $a^2 + b^2 = 1$. Further, let $n_1 = (1, c, d)^T$ and $n_2 = (1, c', e)^T$, (the scale of the normals is not relevant). Given the active normals, the translation is given by $t = n_1 \times n_2$ (Figure 5). For

a planar motion this should be perpendicular to the z -axis. This gives us our last condition,

$$\det (e_z \ n_1 \ n_2) = 0.$$

The determinant condition turns into $c' = c$. This also means that

$$Rn_1 = \begin{pmatrix} a - bc \\ b - ac \\ d \end{pmatrix} \text{ and } Rn_2 = \begin{pmatrix} a - bc \\ b - ac \\ e \end{pmatrix}.$$

Hence we can replace the rotation parameters with $f = a - bc$ and $g = b - ac$. That this should be a rotated version of the first two rows of n_1 can be enforced by $f^2 + g^2 = 1 + c^2$. In summary,

$$\begin{aligned} (1 \ c \ d) N'_1 (1 \ c \ d)^T &= 0, (f \ g \ d) N_1 (f \ g \ d)^T = 0, \\ (1 \ c \ e) N'_2 (1 \ c \ e)^T &= 0, (f \ g \ e) N_2 (f \ g \ e)^T = 0, \\ \text{and } f^2 + g^2 &= 1 + c^2. \end{aligned}$$

There are 5 unknowns and 5 quadratic equations, so there may be up to 32 solutions. In general, Condition 1 will be fulfilled.

8.1 Experimental Validation

In this case we collected a set of 75 view pairs and established point correspondences with SIFT, selecting 10 correspondences for each view pair at random. For

	Error threshold	
	5 pixels	8 pixels
Same result	57	60
1 outlier less	24	20
2 outliers less	2	3

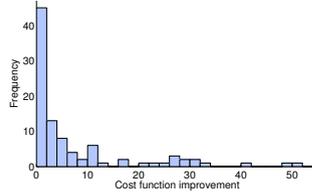


Fig. 6. (Planar motion). (a) Improvement in minimizing the number of outliers. (b) Improvement for the truncated L_2 norm.

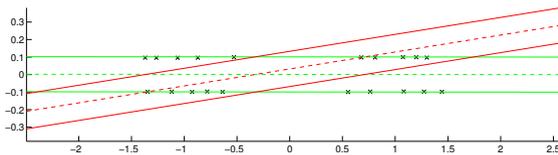


Fig. 7. The red lines show the best possible standard solution (dashed) with error margins (solid), which has 10 outliers and 10 inliers. The green lines show the optimal solution which has no outliers.

the comparisons we used the two-point solver proposed in [16]. Figure 6 shows the results for the two cost functions in Figure 2. The table in Figure 6(a) shows the improvement over best possible RANSAC solution for two different thresholds. Figure 6(b) shows a histogram over the improvement for the truncated L_2 norm, truncated at 3 pixels and approximated with three steps at 1, 2 and 3 pixels. The improvement is in terms of the sum of squared truncated errors, so the unit is pixels². Note that the exact loss function is used for the comparison and not the approximated version.

Since we have as many as 5 quadratic equations, the generic solver required 0.35 s per case. A specialized solver for similar problems runs in about 10 ms.

9 Concluding Discussion

We have presented a framework for computing optimal solutions to multiple view geometry problems with robust error norms. In this section, we will discuss some limitations, how to deal with higher dimensional models, and how it would be possible to speed up the computations. Another possible extension is to fit multiple models and we will briefly discuss how this can be done.

A pathological example. The experiments show that using the standard zero-error minimal solver indeed yields poorer results than the proposed method. However, the difference is typically small. So how bad can it get? Will the standard solver always perform reasonably well? Fig. 7 shows the contrary. For this line-fitting example the standard solver finds a solution with 50% outliers compared to the optimal outlier-free solution.

Higher dimensions. All the theoretical results in this paper are stated for d -dimensional models, but the exhaustive search algorithm quickly becomes impractical for large d . Still, all hope is not lost. A possibility is to apply a random sampling strategy, similar to RANSAC. Another alternative is to use some other pruning technique for some of the dimensions, and then our robust search method for the other ones. For example, in [17], a branch-and-bound technique over $SO(3)$ is proposed for estimating relative orientation (without any outlier rejection). In each bounding step, one needs to solve for the translation component which has two degrees of freedom. This step can be replaced by the method proposed here. This would lead to an algorithm that computes a global optimal solution with a robust error norm.

Speed-ups. In our current MATLAB implementation, we do not use a specialized polynomial equation solver. Replacing this component with a state-of-the-art method such as [15] would speed up the computations significantly as each minimal problem typically can be solved in a couple of milliseconds. Another easily accomplished speed-up would be to do the computations in parallel. Each minimal problem can be solved independently from the others.

Multiple models. If one were to compute multiple models as, for example, in motion segmentation, a greedy approach is often used. This might lead to far from optimal results. However, with similar techniques as developed in this paper, it is actually possible to find optimal solutions in the multi-model case as well. One can show that the optimal N -model solution consists of N models from the set of hypotheses generated by the single-model case.

Acknowledgments. This work was funded by the Swedish Foundation for Strategic Research through the programmes *Future Research Leaders* and *Wearable Visual Information Systems* and by the European Research Council through the programme *GlobalVision*.

References

1. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.* (1981)
2. Chum, O., Matas, J.: Optimal randomized ransac. *Trans. Pattern Analysis and Machine Intelligence* (2008)
3. Hartley, R., Sturm, P.: Triangulation. *Computer Vision and Image Understanding* (1997)
4. Kahl, F., Hartley, R.: Multiple view geometry under the L_∞ -norm. *Trans. Pattern Analysis and Machine Intelligence* (2008)
5. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. *Trans. Pattern Analysis and Machine Intelligence* (2007)
6. Sim, K., Hartley, R.: Removing outliers using the L_∞ -norm. In: *Conf. Computer Vision and Pattern Recognition* (2006)
7. Olsson, C., Eriksson, A., Hartley, R.: Outlier removal using duality. In: *Conf. Computer Vision and Pattern Recognition* (2010)
8. Yu, J., Eriksson, A., Chin, T.J., Suter, D.: An adversarial optimization approach to efficient outlier removal. In: *Int. Conf. Computer Vision* (2011)
9. Breuel, T.: Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding* (2003)
10. Li, H.: Consensus set maximization with guaranteed global optimality for robust geometry estimation. In: *Int. Conf. Computer Vision* (2009)
11. Cass, T.: Polynomial-time geometric matching for object recognition. *Int. Journal of Computer Vision* (1999)
12. Li, H.: A practical algorithm for L_∞ triangulation with outliers. In: *Conf. Computer Vision and Pattern Recognition* (2007)
13. Olsson, C., Enqvist, O., Kahl, F.: A polynomial-time bound for matching and registration with outliers. In: *Conf. Computer Vision and Pattern Recognition* (2008)
14. Bazaraa, M., Sherali, H., Shetty, C.: *Nonlinear Programming: Theory and Algorithms*. Wiley (1993)
15. Byröd, M., Josephson, K., Åström, K.: Fast and stable polynomial equation solving and its application to computer vision. *Int. Journal of Computer Vision* (2009)
16. Källén, H., Ardö, H., Enqvist, O.: Tracking and reconstruction of vehicles for accurate position estimation. In: *W. on Applications of Computer Vision* (2011)
17. Hartley, R., Kahl, F.: Global optimization through rotation space search. *Int. Journal of Computer Vision* (2009)