A Polynomial-Time Bound for Matching and Registration with Outliers

Carl Olsson
calle@maths.lth.se

Olof Enqvist
olofe@maths.lth.se

Fredrik Kahl
fredrik@maths.lth.se

Centre for Mathematical Sciences
Lund University, Sweden

Abstract

We present a framework for computing optimal transformations, aligning one point set to another, in the presence of outliers. Example applications include shape matching and registration (using, for example, similarity, affine or projective transformations) as well as multiview reconstruction problems (triangulation, camera pose etc.).

While standard methods like RANSAC essentially use heuristics to cope with outliers, we seek to find the largest possible subset of consistent correspondences and the globally optimal transformation aligning the point sets. Based on theory from computational geometry, we show that this is indeed possible to accomplish in polynomial-time. We develop several algorithms which make efficient use of convex programming. The scheme has been tested and evaluated on both synthetic and real data for several applications. [1]

1. Introduction

The alignment of point sets is a core problem in computer vision. It appears as a subroutine in many application problems, such as shape matching [3], location matching [21], matching widely separated views [23], shape registration [4] and image registration, see [24] and the references therein. Another, perhaps less obvious, set of problems that can be cast as aligning point sets comes from multiple view reconstruction problems [8]. Applications include triangulation, camera resectioning (pose) and homography estimation to name a few. In this paper, all these matching, registration and reconstruction problems are adressed as alignment problems in a common framework.

In particular, we focus on obtaining globally optimal transformations in the presence of outliers. The problem is posed as finding the largest, consistent subset of all hypothetical correspondences and define the remaining set of correspondences as outliers. A naive way of solving this problem would be to examine all possible subsets of the correspondence set. This is intractable due to the combinatorial explosion of subsets. Our main theorem gives a polynomial upper bound on the number of subsets that need to be examined, expressed as a function of the size of the correspondence set. Unfortunately, this bound is not very useful in practice since there are still far too many subsets to examine for most real world applications. However, the result gives valuable theoretical insights. It helps us to understand why RANSAC is so successful for many applications. Also, it has given us guidelines for designing an alternative approach. By a slight modification of the RANSAC procedure - inspired by the main theorem - we are able to generate high-quality candidates. In turn, given a candidate solution, we present a procedure to verify whether this solution is optimal or not. This verification procedure can be seen as a guided search; either the candidate is verified or a better solution can be found.

In summary, our main contribution is that many matching, registration and reconstruction problems can be solved in polynomial-time with a guarantee of global optimality. We derive practical algorithms for simultaneously (i) computing the optimal transformation and (ii) separating inliers from outliers. There are some limitations to our current framework. We require that the residual errors as functions of the unknown transformation variables are pseudoconvex [2]. This is a weaker condition than convexity (hence convex functions work fine), but it excludes, e.g., 3D-3D registration of point sets with a rigid transformation. Another weakness is that it becomes computationally expensive to obtain global solutions when there is a large portion of outliers, say more than 80%. Typically, solutions with more than 50% inliers are sought for, and if there are no such solutions, then the method is able to report this.

1.1. Related Work

There is a large body of work for solving matching and registration problems [4, 3, 21, 23, 24]. For example, in [3, 12], the matching problem is formulated as an integer program and then solved by non-optimal methods. In [16],

[1] This work has been funded by the European Commission's Sixth Framework Programme (SMErobot grant no. 011838), by the European Research Council (GlobalVision grant no. 209480) and by the Swedish Research Council (grants no. 2004-4579, no. 2005-3230 and no. 2007-6476).

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

the correspondence problem is cast as an assignment problem and can thus be solved optimally. Matching problems have also been solved using ideas based on the Hough transform and Branch-and-bound [18, 5, 9]. Perhaps the most popular paradigm is to use RANSAC [6] which has proven to be successful in many practical situations. Our approach leverages on this idea to generate good candidates for global solutions. However, there are important differences. First of all, RANSAC gives no guarantee of optimality. Then, another problem is that if RANSAC returns an unsatisfactory result, then it is unclear whether there is no better solution or whether RANSAC was unable to find it. With our scheme, we can provide a certificate that there is no subset with, say, more than 50% inliers.

Recently, there has been a renewed interest in multiple view geometry problems aimed at optimal algorithms, e.g., [7, 10, 11]. Using the max-norm, it has been shown that many such problems can be efficiently solved using second order cone programming. The problem of outliers has also been addressed in this context. In [11], a heuristic, non-optimal method is used to remove outliers. In [22] it is shown how to detect *potential* outliers in the optimization process.

The most closely related work is [14] which can be seen as a refinement of [22] as actual outliers are detected and removed. This algorithm, originally introduced in [17], is complementary to our approach and can also be used for computing optimal solutions. Still, our work differs in severals aspects. First, the work of [14] is specialized to the case of triangulation with outliers. Second, the problem formulation is slightly different. While we seek the largest subset of inliers given a preset tolerance, in [14], one has to specify in advance the number of outliers to be removed. This number is in general unknown. Finally, and most importantly, the approach is only practical for problems with *a small number of outliers* and *a small number of unknown variables*, like triangulation (3 variables). In the experimental section, a comparison with our approach is given.

In computational geometry, there is a long tradition of providing performance bounds for different types of geometric optimization algorithms, see [1] for a survey. We have been inspired by this in our work and some terminology is borrowed from that research community.

2. Problem Formulation

Suppose we are given a set of source points in $\mathbb{R}^p$ and a set of target points in $\mathbb{R}^q$ as well as hypothetical correspondences (or pairings) between source and target points. Let H denote the set of correspondences. By an appropriate numbering, corresponding points are given by (x_i, y_i) where $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}^q$ for $i = 1, \dots, m$, where m is the number of pairings, $m = |H|$. A transformation $T : \mathbb{R}^p \mapsto \mathbb{R}^q$ mapping source points x_i to target points $T(x_i)$ is said to be consistent with tolerance α if $d(T(x_i), y_i) \leq \alpha$ for all $i \in I$. Here $d(\cdot, \cdot)$ denotes a metric.

The goal is to find the largest possible subset $I \subseteq H$ under the constraint that there exists a consistent transformation $T \in \mathcal{T}$ for I with a preset tolerance α, where $\mathcal{T}$ is some fixed set of transformations. See Figure 1 for an example. Mathematically, we are interested in solving the following problem,

$$\begin{aligned} \max_{I \subseteq H, T \in \mathcal{T}} \quad & |I| \\ \text{s.t.} \quad & d(T(x_i), y_i) \leq \alpha \quad \forall i \in I. \end{aligned} \tag{1}$$

In addition, for a fixed $I \subseteq H$, we require that the constraints $d(T(x_i), y_i) \leq \alpha$ can be formulated as a convex feasibility problem. (The exact technical assumptions are given in Section 3.2). This allows us to determine whether or not there exists a transformation $T \in \mathcal{T}$ which is consistent with the subset I by convex programming. Using bisection search, one can also find the minimum possible tolerance for I by solving a series of convex feasibility problems, hence computing

$$\alpha^*(I) = \min_{T \in \mathcal{T}} \max_{i \in I} d(T(x_i), y_i), \tag{2}$$

where $\alpha^*(I)$ is the optimal tolerance value. This min-max problem will play a key role in the analysis of the problem. For the applications we consider, an optimal transformation generally exists and is unique. Let T_I^* denote the optimal transformation for I.

At a first glance, the problem in (1) may seem intractable since it is a mixed combinatorial problem - it involves both a combinatorial decision (subset selection) and continuous variables (for the transformation). Using the notation T_I^*, a simple reformulation of (1) is readily available,

$$\begin{aligned} \max_{I \subseteq H} \quad & |I| \\ \text{s.t.} \quad & d(T_I^*(x_i), y_i) \leq \alpha \quad \forall i \in I. \end{aligned} \tag{3}$$

This formulation gives a hint of a possible algorithm: Perform an exhaustive search over all subsets of H and for each subset compute the optimal transformation and verify whether it is consistent or not. However, the complexity of the algorithm is exponential and whence not very useful.

Next we give a few examples that fit into our framework.

Example 2.1. 2D Similarity transformation. The source points x_i are mapped to the target points y_i by a similarity transformation $Rx_i + t$. The rotation and scaling R can be parameterized linearly as

$$R(a,b) = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}. \tag{4}$$

As residual error we use the standard 2-norm $d(Rx_i + t, y_i) = ||Rx_i + t - y_i||$ which is convex in the variables

(a, b, t). Figure 1 shows an instance of the problem without any outliers (o-source points, *-target points). In the upper left figure, the two point sets and their correspondences are shown, and in the upper right the optimal registration. In the lower left, the error residuals are shown. In this case there are 5 residuals that attain the maximum error. To the right is a histogram of the number of residuals obtaining the maximum error when running 1000 repetitions of the problem. Note that we always have 5 or less residuals that attain the maximum error. We shall see in Section 3 that this is no coincident.

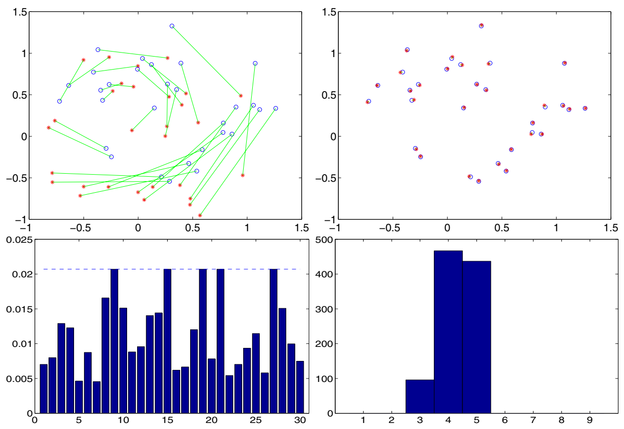

Figure 1. 2D-similarity transformation. See Example 2.1 for details.

Example 2.2. Homography estimation. There are many vision applications involving homographies, for example, detecting a planar configuration viewed in two images. The problem is to determine the homography in the presence of outliers. Let A be a the matrix representing the homography, and A_i be the i'th row of A. We can normalize by letting $A_{33} = 1$. If $x = (x_1, x_2, 1)^T$ and $y = (y_1, y_2, 1)^T$ are two corresponding image points then the error constraint can be written

$$\left|\left|\left(x_1 - \frac{A_1 y}{A_3 y}, x_2 - \frac{A_2 y}{A_3 y}\right)\right|\right| \leq \alpha. \qquad (5)$$

Multiplying the inequality with $A_3 y$ (required to be positive), a convex cone constraint is obtained [10, 11].

This example can easily be generalized to projective transformations and projections for source points in $\mathbb{P}^p$ and target points in $\mathbb{P}^q$. An important case is $(p = 3, q = 2)$ which is the problem of camera pose.

Example 2.3. Triangulation. In this case we have image point measurements and want to reconstruct the 3D-points. Strictly speaking we are not optimizing over a set of transformations but rather the 3D-points. However in our framework we view the image points as target points, the cameras as source points and the 3D-points as transformations. If P is the camera matrix, $u = (u_1, u_2, 1)^T$ the image measurement and $U = (U_1, U_2, U_3, 1)^T$ the corresponding 3D-point then

$$\left|\left|\left(u_1 - \frac{P_1 U}{P_3 U}, u_2 - \frac{P_2 U}{P_3 U}\right)\right|\right| \leq \alpha. \qquad (6)$$

is the reprojection error constraint, which becomes a convex constraint when multiplied with the denominator $P_3 U$.

3. Theory

Before we continue to analyse the problem formulation in (3), we will introduce some useful concepts from computational geometry (see [1]).

3.1. Bases and Support Sets

If there exists a subset $S \subseteq I$ having the same optimal value $\alpha^*(S) = \alpha^*(I)$, that is,

$$\min_{T \in \mathcal{T}} \max_{i \in I} d(T(x_i), y_i) = \min_{T \in \mathcal{T}} \max_{s \in S} d(T(x_s), y_s), \qquad (7)$$

then we say that S and I are equivalent, and write $S \sim I$.

The elements in I for which $d(T(x_i), y_i) = \alpha^*(I)$ (the active constraints) form the *support set*. By a *basis* B of I we mean a set B that is equivalent to I but contains no equivalent proper subset. Or formally

Definition 3.4. B is a basis set of I if $B \sim I$ and for all $S \subseteq B$, $S \sim I$ we have $S = B$.

The maximum cardinality of any basis is called the *combinatorial dimension* of I.

3.2. Main Theorem

In this section we will show that the combinatorial dimension is always (relatively) small and independent of the cardinality of the constraint set H. Let $t \in \mathbb{R}^n \mapsto T_t \in \mathcal{T}$ be a parameterization of our space of transformations, and $r_i(t) = d(T_t(x_i), y_i)$ be the residual errors.

We will show that if $t \in \mathbb{R}^n$ then the size of the basis sets is always less than or equal to $n + 1$. To do this we will assume that the residuals $r_i(t)$ are pseudoconvex which is a slightly less restrictive assumption than convexity [2]. It was shown in [19] that this is true for a large number of multiview geometry problems. In particular, it is true for all the examples given in Section 2. First, we need the following classical theorem ([20]):

Theorem 3.5 Caratheodory. *Let M be a set of points in $\mathbb{R}^n$ and let $C = conv(M)$. Then $t \in C$ if and only if t can be expressed as a convex combination of $n + 1$ points in M.*

We also need the following theorem from [19], which establishes necessary and sufficient conditions for a global optimum for min-max problems.

Theorem 3.6. *Let $r_i(t)$ be pseudoconvex functions for $i = 1, \ldots, m$. Then, t^* solves $\mu^* = \min_t \max_j r_j(t)$ if and only if there exists λ_i such that*

$$\sum_{j=1}^{m} \lambda_j \nabla r_j(t^*) = 0 \quad \textstyle\sum_{j=1}^{m} \lambda_j = 1, \tag{8}$$

where $\lambda_i \geq 0$ if $r_i(t^*) = \mu^*$ and $\lambda_i = 0$ if $r_i(t^*) < \mu^*$ for $i = 1, \ldots, m$.

These conditions can be interpreted as follows. Consider the point t^* and the gradients $\nabla r_i(t^*)$ for which $r_i(t^*) = \max_j r_j(t^*)$. The point t^* is a global minimum if the origin is a positive linear combination of these gradients. Geometrically this means that in each direction there is at least one residual that grows. In general these conditions only ensure a local minimum, but when $r_i(t)$ are pseudoconvex, they give the global minimum.

We are now ready to prove our main theorem.

Theorem 3.7. *Let (t^*, μ^*) be the optimal solution of $\mu^* = \min_{t \in \mathbb{R}^n} \max_{i \in I} r_i(t)$ for some set I. Then, if $B \sim I$ and B is a basis of I then $|B| \leq n + 1$.*

Proof. There are $\lambda_i \geq 0, i = 1, \ldots, |I|$ such that the conditions of Theorem 3.6 are fulfilled. We assume that $\lambda_{i_k} > 0$, $k = 1, ..., l$ where $l > n + 1$, otherwise we are done.

Let M be the set $\{\nabla r_{i_1}(t^*), ..., \nabla r_{i_l}(t^*)\}$. Since $\lambda_i \geq 0$, conditions (8) imply that $0 \in \text{convhull}(M)$. By Caratheodory's theorem, 0 can be written as a convex combination of $n + 1$ points from M. Hence there exist $\tilde{\lambda}_i$ such that conditions (8) are fulfilled with $\tilde{\lambda}_i \neq 0$ for at most $n+1$ of $\tilde{\lambda}_i$. Next, letting $B = \{\, i\, ;\, \tilde{\lambda}_i > 0 \,\}$ it is easy to see that the conditions of Theorem 3.6 are fulfilled and thus $B \sim I$. □

Note that the combinatorial dimension is independent of the cardinality of the constraint set H.

3.3. A Reformulation

According to Theorem 3.7, it is enough to consider basis sets. We get the following equivalent reformulation of (3) in the sense that the formulations have the same optimum.

$$\begin{aligned} \max_{I \subseteq H} \quad & |I| \\ \text{s.t.} \quad & d(T^*_B(x_i), y_i) \leq \alpha \quad \forall i \in I \\ & B \text{ is a basis for } I. \end{aligned} \tag{9}$$

This leads to the following algorithm.

Algorithm 3.8. For each basis B:

- Determine the transformation T^*_B by solving $\arg\min_{t \in \mathbb{R}^n} \max_{i \in B} r_i(t)$.
- Determine the number of inliers for this transformation by evaluating $d(T^*_B(x_i), y_i) \leq \alpha$ for all $i \in H$.

Assume that the number of residuals is m. Since the size of the basis sets are bounded by $n + 1$ we know that it is enough to test every possible subset of $n + 1$ residuals. The number of such sets is $O(m^{n+1})$. Since step 1 is $O(f(n))$ for some function $f(n)$ and step 2 is $O(m)$ the above search scheme is polynomial in the problem size m. Although the result is encouraging, the running time is still too large to be practical in many cases. Consider for instance the camera pose problem with outliers. Here the transformation (the camera matrix) has 11 degrees of freedom and therefore we obtain an algorithm that is $O(m^{13})$.

A more practical way to capitalize on this knowledge is to use basis sets in RANSAC. The standard approach is to apply minimal sets, but as the following example shows, there may be no minimal set that finds the global optimum. Consider the problem of fitting two point sets with known correspondences using a similarity transformation (Figure 2). In the upper left figure, the two point sets (red and blue) and their correspondences are shown. If we choose α to be 0.05 it is possible to find two solutions with 3 inliers using a minimal solver. The two initial solutions are shown in the upper right and lower left figures, respectively. In both cases the 4 points to the left are selected as outliers. However, the optimal solution has the 4 points to the left as inliers, as shown in the bottom right figure. Hence when using minimal solvers there are cases where it is not possible to find a starting point that can be improved into the global maximum.

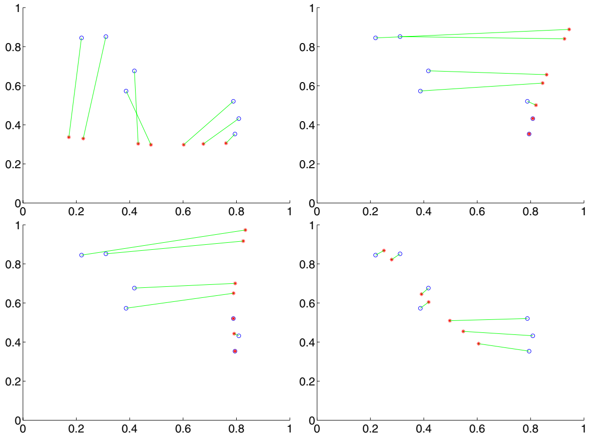

Figure 2. An example where standard RANSAC will fail.

Since the combinatorial dimension of this problem is 5, we know that by selecting 5 points instead of 2 and solving the min-max problem it is always possible to find the correct transformation, in this case it suffices to pick 3 points however. The price we pay for this improvement is that the probability of picking only inliers decreases when we select larger sets. Also the solvers for overdetermined systems are in general slower than minimal solvers.

4. Algorithms

As mentioned in the previous section the optimal solution to (1) can be found by performing an exhaustive search over all possible basis sets. Since this is seldom computationally feasible, we consider a dual approach to the problem. Rather than trying to solve a difficult general problem globally, we start with a local optimum and try to verify that it is in fact global.

We return to the much used RANSAC approach. Its wide use in a variety of applications tells us that the method is often effective in finding a good solution. The big drawback is that we never get any guarantee of global optimality. Thus we suggest using a RANSAC-type algorithm to generate a candidate solution, and then a second step to verify the global optimality of this solution or to find a better candidate solution. Another advantage is that in cases where no satisfactory candidate solution is obtained, we can prove that this must be the case, that is, we get a hard upper bound on the number of inliers for any solution.

4.1. Finding Local Maxima

We use the standard RANSAC algorithm to generate a first candidate solution I consistent with some transformation T. Then we try adding more correspondences from $H \setminus I$ to this set. When no more correspondences can be added without violating consistency, we say that we have reached *a local maximum*. In order to verify that a solution set I is indeed a local maximum, one needs to perform one feasibility test of the set $B \cup \{i\}$ for each $i \in H \setminus I$ where B is a basis of I.

In most cases, we can use minimal subsets in the RANSAC algorithm, or any local algorithm, to generate the first candidate solution. For hard problems, it might be advantageous to use subset size equal to the combinatorial dimension as described in Section 3.3.

4.2. Finding Global Maxima

After termination of the local maximum search, we will be in one of the following situations.

- The globally optimal solution has been found and it has a high rate of inliers.
- There is no candidate solution with a satisfactorily high rate of inliers.
- The global optimum has a high rate of inliers but it has not been found yet.

In the next section we will describe an algorithm to verify that the solution we have found is in fact the global optimum, if this is true, and show how the same algorithm can be used in the case where no satisfactory candidate solution has been found.

4.2.1 Optimality Verification

Starting from our set of hypothetical correspondences, H, assume that we have found a local maximum, i.e., a set $I_l \subset H$ consistent (within a given tolerance α) with a transformation T_l. Since it is a local maximum, no more correspondences can be added to I_l. We wish to prove that I_l is optimal in the sense of (1) or find a better candidate set. To verify that we have found the global optimum, we need to reject the hypothesis that there exists a set $I^* \neq I_l$ consistent with some transformation T^* such that $|I^*| > |I_l|$.

We call a correspondence $i \notin I_l$ that can not be added to I_l without picking a larger α a l-outlier. We first make the (trivial) observation that I^* would have to contain at least one l-outlier. Our strategy to prove optimality of I_l is to show, for each l-outlier, that it cannot lie in I^*.

Algorithm 4.9. Discarding one l-outlier from I^*.

- Pick an l-outlier i. Divide the set $H \setminus \{i\}$ into disjoint sets H_k of n or more elements each (where $n+1$ is the combinatorial dimension of the problem).
- For each set $H_k \cup \{i\}$, perform a feasibility test with tolerance α. If the test fails then we say that i is inconsistent with H_k.
- If the total number of sets that are inconsistent with i is larger than the number of l-outliers we can remove i completely from the problem and continue our iteration with one outlier less (i.e., set $H := H \setminus \{i\}$).

The last step can be motivated as follows. If a set H_k is not consistent with i, then $i \in I^*$ implies that at least one correspondence from H_k is not in I^*. Thus, if $|H \setminus I_l| + 1$ of the sets H_k are inconsistent with i then $i \in I^*$ implies $|I^*| < |I_l|$. Hence we can conclude that $i \notin I^*$.

It is easy to calculate the maximum rate of outliers that we can handle with the algorithm described above. We get $\frac{|H|-1}{n}$ sets H_k in Algorithm 4.9, so we can verify I_l with at most $\frac{|H|-1}{n} - 1$ outliers. If H is large the rate of outliers we can handle is approximately $\frac{1}{n}$.

Naturally we would like to improve this performance without having to do an exhaustive search. This can be done by choosing non disjoint sets. In this way it is possible to generate more test sets. As the number of outliers grows the problem becomes more difficult until eventually it requires testing all sets of size $n+1$, resulting in exhaustive basis search.

A rather general method is the following modification of the approach in Algorithm 4.9. Divide H into disjoint sets H_k of size $n+d$ (instead of n). For each such set form all subsets of size n, denoted $H_{k,j}$. Now we check which of the sets $H_{k,j}$ that are consistent with i. The results tell us something about how many elements of H_k must be $*$-outliers provided that $i \in I^*$.

- If no subsets of H_k are consistent with i at least $d+1$ elements of H_k must be $*$-outliers provided $i \in I^*$.
- If less than $\binom{n+f}{f}$ of the sets are consistent at least d+1-f of the elements of H_k must be $*$-outliers provided $i \in I^*$.
- If all subsets of H_k are consistent with i we get no information.

It should be added that the (smallest) subset size does not have to equal the combinatorial dimension. Sometimes it is preferable to choose a larger number to increase the chance of inconsistency.

Remark. It is not necessary to have a candidate solution to start off with. The same algorithm can still be applied to prove a statement of the type $|I^*| < \gamma|H|$ for some $0 < \gamma < 1$. In this case we pick points one by one and show that they cannot be part of any I with $|I| \geq \gamma|H|$.

4.2.2 Finding a better transformation

As already commented there is a risk that the RANSAC-type algorithm used to seek local maxima does not find the globally optimal transformation. In this case the verification will fail to discard all outliers, but it will still give us valuable information. For each outlier that could not be discarded, we will get a number of sets that are not consistent with this outlier and a number of sets that are. This inspires a reverse variant of the verification algorithm.

Algorithm 4.10. Pick an l-outlier that could not be discarded and a set of points that were consistent with this outlier. Use all these points to calculate a new candidate solution. Iterate.

5. Experiments

We have tested our approach on a number of different computer vision problems. In all cases, [15] was used to extract SIFT descriptors from the images and to compute the hypothetical correspondence set H. Then RANSAC was applied to generate candidate solutions. If possible these were verified as global maxima using Algorithm 4.9, but in many cases, the initial solution first had to be improved using either local search (Section 4.1) or Algorithm 4.10. Our implementation was done in Matlab using SeDuMi for feasibility tests.

For each of the experiments, our results are compared to the performance of a standard RANSAC procedure. In the few cases where possible (within reasonable time) we also compare with the algorithm given in [14]. When this is not possible we give a worst case bound on the number of bases from [17]. The results are summarized in Table 1. The tolerance α was set to two pixels.

Homography. Given image correspondences of 3D points lying in a plane, there is a homography between the two image planes mapping corresponding points to each other. Other correspondences can be regarded as outliers.

We tested our algorithm on two stereo image pairs with different outlier rates (see Table 1). Figure 3 shows the computed inliers (green) and outliers (red) for the first stereo pair. In this case the number of inliers are 432 and the number of outliers are 81. The solution was verified to be optimal (see Table 1).

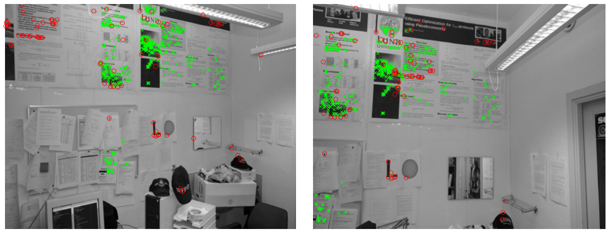

Figure 3. The homography estimation problem with outliers. Inliers (green points) and outliers (red points).

3D Pose. In this experiment we try to determine the camera pose of a 3D object from image data (see Figure 4). We used images from the publicly available database introduced in [13]. The model, plotted in the left of Figure 4), was created from two stereo images of the front of the Teddy bear. Using SIFT correspondences we then estimated the pose in 10 test images where the Teddy is partly visible. In all cases we where able to verify optimality of our local solution (see Table 1).

2D Pose. In this experiment we took 6 images of books on a table, and tried to find the optimal similarity transformation between a pair of images. In one case we matched a model image with only one book to a more complex image with the same book appearing several times. We also tried matching the more complex images to each other. Interestingly, we were able to verify the optimal solution in cases where another strong solution existed, and in cases where no strong solution existed, we were able to prove this (see Figure 5 and Table 1).

Triangulation. We also tested our method on a triangulation example. We took 25 images of a painted vase and matched SIFT features from different views. The camera positions were estimated using markers. Then we tried to verify the solutions obtained by performing RANSAC on the correspondence data. There were 269 points that had been matched to at least seven images. Of these, we successfully verified 190 solutions to be optimal that had at least 50% inliers. In the remaining 79 cases we could verify that there was no solution with 50% inliers or more.

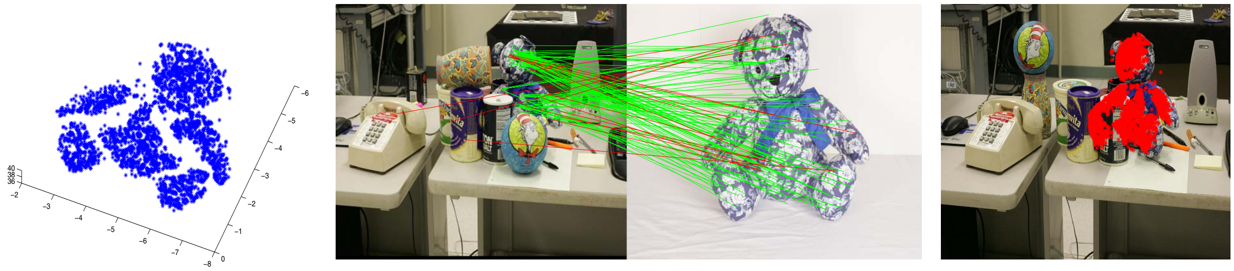

Figure 4. The 3D pose problem. Left: The model obtained from two images. Middle: Correspondences between the test image and one of the stereo images (green - outliers, red - inliers). Right: The model points projected onto the test image under the optimal projection.

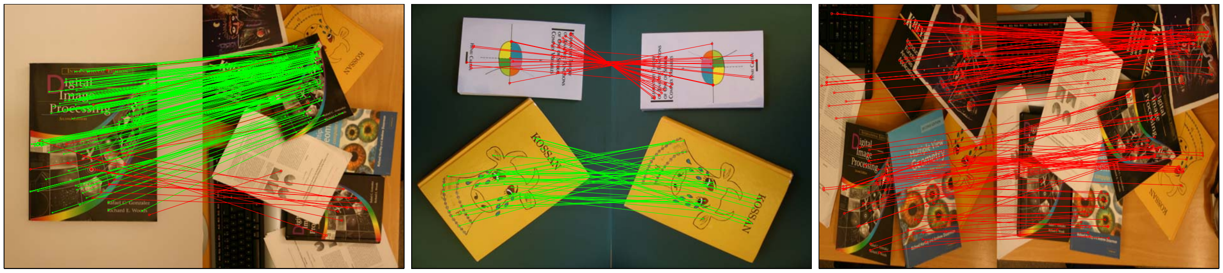

Figure 5. Images of books used for estimating similarity transformations. Green points are inliers of the globally optimal transformation and red rings are outliers. The middle images show that we can verify the optimal transformation despite the presence of another strong transformation, and to the right we have an example of a case where we can verify that there is no transformation with at least 50 % inliers.

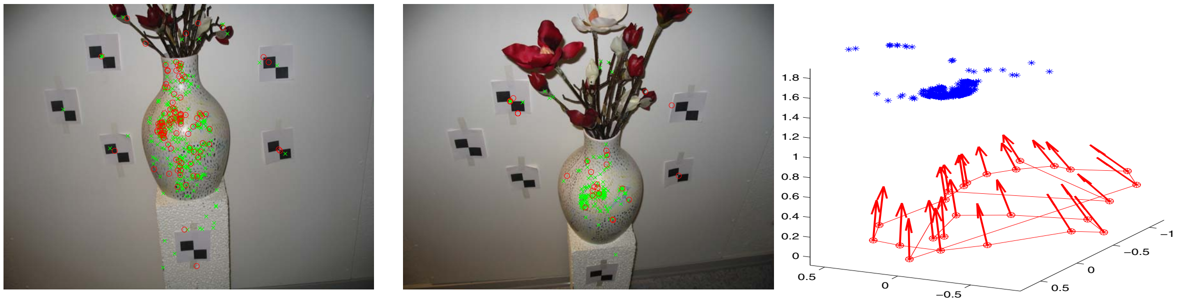

Figure 6. The triangulation problem. Two images (out of 25) with points used for calculating the reconstruction to the right. Green points (inliers) are used for the triangulation while red points are detected as outliers because of miss matchings.

6. Conclusions

In contrast to most previous work dealing with outliers, we have presented a framework for estimating globally optimal solutions. For a large number of applications, we have shown both theoretically and experimentally that this is indeed a tractable problem. In all experiments we have been able to either produce the optimal solution or prove that there is no solution with more than 50% inliers.

Another conclusion of this paper is that RANSAC works rather well. We have given both theoretical evidence and practical experiments which show that it is a sound method. The framework presented in this paper can also be used for benchmarking other local methods dealing with outliers.

References

[1] P. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30(4):412–458, 1998.

[2] Bazaraa, Sherali, and Shetty. *Nonlinear Programming, Theory and Algorithms.* Wiley, 1993.

[3] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 26–33, San Diego, USA, 2005.

[4] P. Besl and N. McKay. A method for registration two 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] T. Cass. Polynomial-time geometric matching for object recognition. *Int. Journal Computer Vision*, 21(1–2):37–61, 1999.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–395, 1981.

[7] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 504–509. 2004.

	Image id	Correspondences $|H|$	Inliers RANSAC	Inliers Local method	Number of tests to verify optimum	Number of tests in [14] (* = worst case bound)
Homography (Office Wall)	1	513	430	432	5889	$6.7 \cdot 10^{15*}$
	2	101	57	64	9468	$4.6 \cdot 10^{13*}$
3D Pose (Teddy Bear)	1	353	264	270	10276	$4.0 \cdot 10^{21*}$
	2	121	104	105	420	$9.3 \cdot 10^{13*}$
	3	69	49	50	1171	$5.6 \cdot 10^{14*}$
	4	86	55	57	800	$5.6 \cdot 10^{14*}$
	5	150	114	116	834	$6.5 \cdot 10^{15*}$
	6	65	42	50	1228	$4.8 \cdot 10^{13*}$
	7	14	9	9	1001	$2.7 \cdot 10^{5*}$
	8	105	87	87	418	$3.2 \cdot 10^{14*}$
	9	174	147	149	718	$1.0 \cdot 10^{16*}$
	10	263	244	245	187	$3.2 \cdot 10^{14*}$
2D Pose (Books)	1	217	194	199	396	$3.5 \cdot 10^{5*}$
	2	67	56	59	102	$4.9 \cdot 10^{5*}$
	3	76	70	71	20	574
	4	74	66	66	42	11177
	5	77	46	46	4563	$2.8 \cdot 10^{6*}$
	6	146	43	< 73	18335	$3.2 \cdot 10^{8*}$

Table 1. *Summary of experimental results. Correspondences $|H|$ - the total number of hypothetical correspondences obtained from the SIFT descriptors. Inliers - the number of inliers detected by standard RANSAC and the local search method, respectively. Number of tests to verify optimum - number of convex feasibility tests required for verifying that the local maximum (output of the local method) is global. Number of tests in [14] - number of bisection programs required to discard the same number of outliers as the local method using the method from [14]. When it is not possible to run in practice a worst case bound from [17] is given. Note that a bisection program consists of approximately 10-15 feasibility tests. All solutions except 2D pose image id 6 were verified to be optimal. In this case we verified that there was no solution with more than* 50% *inliers.*

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

[9] D. Jacobs. Matching 3-d models to 2-d images. *Int. Journal Computer Vision*, 21(1–2):123–153, 1999.

[10] F. Kahl. Multiple view geometry and the L_∞-norm. In *International Conference on Computer Vision*, pages 1002–1009. Beijing, China, 2005.

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

[12] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Solving markov random fields using second order cone programming relaxations. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 1045–1052, New York City, USA, 2006.

[13] A. Kushal and J. Ponce. Modeling 3d objects from stereo views and recognizing them in photographs. *Proc. European Conference on Computer Vision*, 2006.

[14] H. Li. A practical algorithm for L_∞ triangulation with outliers. In *Proc. Conf. Computer Vision and Pattern Recognition*. Minneapolis, USA, 2007.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal Computer Vision*, 2004.

[16] J. Maciel and J. Costeira. A global solution to sparse correspondence problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.

[17] J. Matousek. On the geometric optimization with few violated constraints. *Symposium on Computationsl Geometry*, pages 312–321, 1994.

[18] C. Olson. A general method for geometric feature matching and model extraction. *Int. Journal Computer Vision*, 45(1):39–54, 2001.

[19] C. Olsson, A. Eriksson, and F. Kahl. Efficient optimization for L_∞-problems using pseudoconvexity. *Proc. International Conference on Computer Vision*, 2007.

[20] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[21] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92:236–264, 2003.

[22] K. Sim and R. Hartley. Removing outliers using the L_∞-norm. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 485–492. New York City, USA, 2006.

[23] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *Int. Journal Computer Vision*, 59(1):61–85, 2004.

[24] G. Yang, C. Stewart, M. Sofka, and C.-L. Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(11):1973–1989, 2007.