

## Fast and Reliable Two-View Translation Estimation

Johan Fredriksson<sup>1</sup>

<sup>1</sup>Centre for Mathematical Sciences  
Lund University, Sweden

johanf@maths.lth.se

Olof Enqvist<sup>2</sup> Fredrik Kahl<sup>1,2</sup>

<sup>2</sup>Department of Signals and Systems  
Chalmers University of Technology, Sweden

{olof.enqvist, fredrik.kahl}@chalmers.se

### Abstract

*It has long been recognized that one of the fundamental difficulties in the estimation of two-view epipolar geometry is the capability of handling outliers. In this paper, we develop a fast and tractable algorithm that maximizes the number of inliers under the assumption of a purely translating camera. Compared to classical random sampling methods, our approach is guaranteed to compute the optimal solution of a cost function based on reprojection errors and it has better time complexity. The performance is in fact independent of the inlier/outlier ratio of the data.*

*This opens up for a more reliable approach to robust ego-motion estimation. Our basic translation estimator can be embedded into a system that computes the full camera rotation. We demonstrate the applicability in several difficult settings with large amounts of outliers. It turns out to be particularly well-suited for small rotations and rotations around a known axis (which is the case for cellular phones where the gravitation axis can be measured). Experimental results show that compared to standard RANSAC methods based on minimal solvers, our algorithm produces more accurate estimates in the presence of large outlier ratios.*

### 1. Introduction

Estimating the epipolar geometry in two views is a classical problem in computer vision. Perhaps the most well-known method is the eight point algorithm by Longuet-Higgins [12], which gained popularity due to its simplicity and efficiency. However, it does not properly handle measurement noise and mismatches among the image correspondences. In this paper, we study the special case of translation estimation in two views. Either the camera is known not to rotate or the relative rotation can be obtained by some other means. So, why is this setting worth studying? First, there are many situations where the camera undergoes a pure translation, for example, in robotic applications. Then, the solution to this problem turns out to be very simple and computationally tractable.

The most closely related work is the paper by Hartley and Kahl [7], where a branch and bound framework is introduced that searches over all rotations. As a subroutine, the translation of the camera is estimated using linear programming. In contrast to our work, it is assumed that no outliers are present in the data. Our basic translation estimator could be used instead in order to obtain an algorithm that computes the globally optimal solution in the presence of outliers. A related branch and bound approach for dealing with outliers is given in [4]. Yet another branch and bound method for the uncalibrated setting based on an algebraic cost function was developed in [11]. A serious drawback of these approaches is that they are based on a computationally costly scheme with unpredictable running times. We will not follow the branch and bound path and limit the scope to restricted motions in this paper.

In multiple view geometry, RANSAC has proved to be a useful technique for dealing with outliers [5, 15, 8]. The approach has also been specialized to the case with rotation around one axis [6, 14]. This is motivated by the fact that many modern cell phones are able to directly measure the gravitation vector. One disadvantage with these methods is the randomness of the results and that the solution space is restricted to the hypotheses given by minimal subsets of the data. Thus, there is no guarantee of actually finding a good solution even if there exists one. A direct application of the general theory of optimal robust model fitting proposed in [3] would yield an  $\mathcal{O}(n^3)$ -algorithm for minimizing the number of outliers. Our approach is significantly faster. Structure from motion with known rotations has previously been considered, first with algebraic cost functions [17], then later with reprojection errors in the  $L_\infty$ -framework of [9]. However, it is assumed that there are no outlier correspondences. Extensions to outliers can only handle limited amounts of outliers [10, 16, 18].

In this paper, we analyze in depth the case of two-view translation estimation. Our main contributions are:

- We develop a provably optimal algorithm that, for a given error threshold  $\epsilon$ , finds a translation which maximizes the number of correspondences with reprojec-

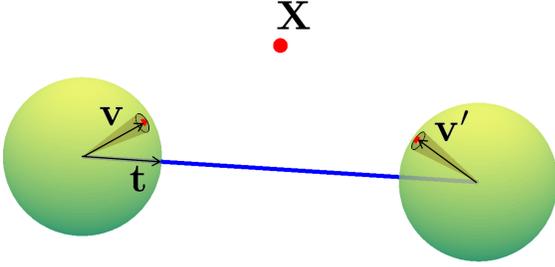


Figure 1: *Epipolar geometry*. The translational direction  $\mathbf{t}$ , from one camera centre to the other, should lie in the same plane as the two image vectors  $\mathbf{v}$  and  $\mathbf{v}'$  of the 3D point  $\mathbf{X}$ .

tion errors below  $\epsilon$ . The time complexity of the algorithm is  $\mathcal{O}(n^2 \log(n))$ , where  $n$  is the number of hypothetical correspondences. Note that performing exhaustive search over all minimal subsets in a RANSAC-manner would result in an algorithm that has complexity  $\mathcal{O}(n^3)$ .

- We experimentally demonstrate that our algorithm performs very well in several realistic and challenging setups including cases with known rotation, known rotation axis and small rotations. The baseline method for our comparisons is RANSAC and our experiments indicate that we cannot only obtain more accurate results, but also can be faster, particularly in presence of large amounts of outliers.

## 2. Translational Epipolar Geometry

We assume calibrated cameras undergoing a purely translational motion. As is commonly done with calibrated cameras, we find it convenient to consider image points as lying on an image sphere, rather than an image plane. Thus, an image measurement is represented by a unit vector giving the direction from the camera centre to the 3D point.

Given a correspondence pair  $(\mathbf{v}, \mathbf{v}')$  in two images viewing a 3D point  $\mathbf{X}$ , the epipolar constraint tells us that the three vectors  $\mathbf{v}$ ,  $\mathbf{v}'$  and  $\mathbf{t}$  should be coplanar as illustrated in Fig. 1. The translation direction  $\mathbf{t}$  from one camera centre to the other can be assumed to have unit norm since the global scale cannot be determined in any case. In practice, one has to take into account measurement noise and ideally, we would like that the reprojections of  $\mathbf{X}$  should lie as close to the measured image points  $\mathbf{v}$  and  $\mathbf{v}'$  as possible. We require that the reprojection is smaller than some predefined angular error  $\epsilon$ , which can be written

$$\angle(\mathbf{v}, \mathbf{X}) \leq \epsilon \quad \text{and} \quad \angle(\mathbf{v}', \mathbf{X} - \mathbf{t}) \leq \epsilon,$$

where  $\angle$  is the angle between  $\mathbf{v}$  and  $\mathbf{v}'$ . The above condi-

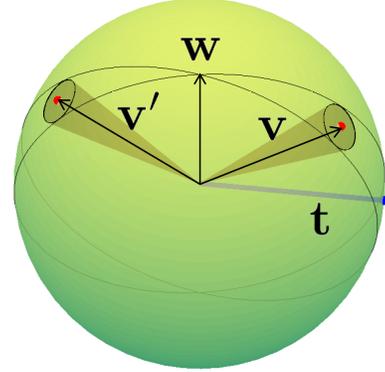


Figure 2: *Epipolar constraint on a sphere*. The translation  $\mathbf{t}$  is bounded to lie in the wedge formed by two great circles of the sphere. The great circles are tangent to the  $\epsilon$ -cones around  $\mathbf{v}$  and  $\mathbf{v}'$ , respectively.

tions can be equivalently expressed as

$$\frac{\|\mathbf{v} \times \mathbf{X}\|}{\mathbf{v} \cdot \mathbf{X}} \leq \tan(\epsilon) \quad \text{and} \quad \frac{\|\mathbf{v}' \times (\mathbf{X} - \mathbf{t})\|}{\mathbf{v}' \cdot (\mathbf{X} - \mathbf{t})} \leq \tan(\epsilon), \quad (1)$$

provided that the denominators are positive. For a fixed  $\mathbf{t}$ , each such condition constrains the 3D point  $\mathbf{X}$  to lie in a convex cone, and hence the feasible set of 3D points satisfying (1) is convex and obtained as the intersection of two cones [9]. We say that a correspondence pair  $(\mathbf{v}, \mathbf{v}')$  is *feasible* for  $\mathbf{t}$  if the feasible set of 3D points is non-empty. This leads us to the following problem formulation.

**Problem 1.** *Given image correspondences  $(\mathbf{v}_i, \mathbf{v}'_i)$ ,  $i = 1, \dots, n$  and an angular error bound  $\epsilon$  in two views, find a translation  $\mathbf{t}$  such that the number of feasible correspondences is maximized.*

It turns out that one can simplify the epipolar constraints of (1) by further analyzing the geometry of the setup. If the two image vectors  $\mathbf{v}$  and  $\mathbf{v}'$  are placed on the same sphere together with the translation vector  $\mathbf{t}$ , then — in the noise-free case — the coplanarity constraint is equivalent to the three vectors lying on a great circle of the sphere. If we now include the angular error bound  $\epsilon$  as previously specified, then we see that the translation is constrained to lie on a wedge bounded by two great circles, cf. Fig. 2. Conversely, for any  $\mathbf{t}$  that lies in such a wedge, there exists a 3D point  $\mathbf{X}$  such that when  $\mathbf{X}$  is reprojected to the image, the angular error between the measured image vector and the reprojection is less than  $\epsilon$ . In other words, the correspondence  $(\mathbf{v}_i, \mathbf{v}'_i)$  is feasible provided its wedge contains  $\mathbf{t}$ . This observation was first made in [7]. Note that only the wedge in Fig. 2 containing  $\mathbf{v}$  will yield a 3D point  $\mathbf{X}$  with positive depths with respect to the two image vectors.

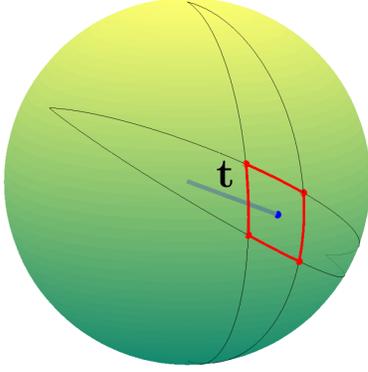


Figure 3: Two correspondences constrain the translation  $\mathbf{t}$  to be in the intersection of two wedges.

Hence, in order for two correspondence pairs to be feasible simultaneously, their corresponding wedges must have a non-empty intersection as illustrated in Fig. 3. Analogously, for a set of correspondences to be feasible for the same translation, the corresponding wedges should have a common intersection point. Inspired by computational geometry terminology [2], the *depth* of a point is defined to be the number of wedges on which the point lies. For example, the vector  $\mathbf{t}$  in Fig. 3 has depth two. Note that the depth is constant on the intersection sets of the wedges. Thus, a set of wedges tessellates the sphere into a set of spherical polygons bounded by great circles and on each such polygon, the depth is constant, see Fig. 4. This leads us to the following reformulation of our problem.

**Problem 2.** *Given a set of  $n$  wedges on the sphere, find a point of maximum depth.*

**Wedge formula.** Consider the pair  $(\mathbf{v}, \mathbf{v}')$  and the angular bound  $\epsilon$ . Let  $\alpha$  be the angle between  $\mathbf{v}$  and  $\mathbf{v}'$  (computed via  $\cos(\alpha) = \mathbf{v} \cdot \mathbf{v}'$ ) and let  $\beta$  be the angle between the two great circles of the wedge at the intersection  $\mathbf{w} = (\mathbf{v} + \mathbf{v}') / \|\mathbf{v} + \mathbf{v}'\|$ , cf. Fig. 2. Then,  $\beta$  can be obtained via the spherical law of sines

$$\sin(\beta/2) = \sin(\epsilon) / \sin(\alpha/2).$$

A great circle can be represented by the normal vector of the plane through the great circle and the origin of the sphere. The plane normal for the great circle through  $\mathbf{v}$  and  $\mathbf{v}'$  is given by  $\mathbf{n} = (\mathbf{v} \times \mathbf{v}') / \|\mathbf{v} \times \mathbf{v}'\|$  and the two normals for the great circles of the wedge are given by

$$\mathbf{n}^\pm = \sin(\beta/2)(\mathbf{n} \times \mathbf{w}) \pm \cos(\beta/2)\mathbf{n}. \quad (2)$$

Throughout the rest of the paper will use the convention that the signs of  $\mathbf{n}^+$  and  $\mathbf{n}^-$  are chosen so that a unit vector  $\mathbf{t}$  lies on the wedge if and only if  $\mathbf{t} \cdot \mathbf{n}^+ \geq 0$  and  $\mathbf{t} \cdot \mathbf{n}^- \geq 0$ .

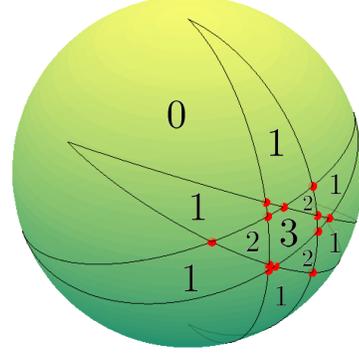


Figure 4: Three wedges with intersection points marked in red. The numbers state the depth of each spherical polygon.

### 3. Finding the Maximum Depth

Consider a set of  $n$  wedges on the sphere represented by pairs of normals  $(\mathbf{n}_i^+, \mathbf{n}_i^-)$ ,  $i = 1, \dots, n$ . A brute-force approach for computing the maximum depth would be to, for each pair of wedges, find the four corner points of the intersecting spherical quadrilateral (Fig. 3) and for each of these points, count how many of the other wedges contain the point. As there are  $\mathcal{O}(n^2)$  wedge pairs, the total complexity of the approach would be  $\mathcal{O}(n^3)$ . This bound can be improved with a more sophisticated scheme.

We will start by describing a procedure for finding the maximum depth along one of the great circles of wedge  $k$ , say  $\mathbf{n}_k^+$ , that starts at point  $\mathbf{w}_k$  and ends at point  $-\mathbf{w}_k$ . This procedure can be applied for all  $2n$  great circles and thereby guaranteeing that the point of maximum depth will be found. First, compute all intersection points for the other wedges ( $i \neq k$ ),

$$\mathbf{t}_i^+ = \frac{\mathbf{n}_i^+ \times \mathbf{n}_k^+}{\|\mathbf{n}_i^+ \times \mathbf{n}_k^+\|} \text{ and } \mathbf{t}_i^- = \frac{\mathbf{n}_k^+ \times \mathbf{n}_i^-}{\|\mathbf{n}_k^+ \times \mathbf{n}_i^-\|}. \quad (3)$$

See Fig. 4. Wedges that are not intersecting the great half circle of  $\mathbf{n}_k^+$  can be discarded. Then, sort the points in the order they appear along the great circle of the sphere. A simple way to do this is to compute the projection of  $\mathbf{t}_i^+$  and  $\mathbf{t}_i^-$  onto  $\mathbf{w}_k$ , that is,

$$\rho_i^+ = \mathbf{t}_i^+ \cdot \mathbf{w}_k \text{ and } \rho_i^- = \mathbf{t}_i^- \cdot \mathbf{w}_k, \quad (4)$$

and then sort the values  $\rho_i^+$  and  $\rho_i^-$ . The next step is to traverse the sorted list. When a point of type  $\mathbf{t}_i^+$  is encountered, a new wedge is entered and the depth is increased by one. Analogously, for a point  $\mathbf{t}_i^-$ , the depth is decreased by one. This will yield all the depths along the great circle and in particular the maximum depth. The complete procedure is summarized in Algorithm 1.

---

**Algorithm 1** Maximum depth

---

Input: Wedge normals  $(\mathbf{n}_i^+, \mathbf{n}_i^-)$ ,  $i = 1, \dots, n$   
Set maxdepth = 0  
For each great circle of a wedge  
  Set depth = 1  
  Compute intersection points with other wedges, cf. (3)  
  Sort the intersections along the great circle, cf. (4)  
  For each intersection in the sorted list  
    If entering a wedge, then  
      depth = depth + 1 and update maxdepth  
    else  
      depth = depth - 1

---

**Complexity.** The sorting step of Algorithm 1 takes  $\mathcal{O}(n \log(n))$  and this has to be repeated  $\mathcal{O}(n)$  times. Hence the total complexity of the algorithm is  $\mathcal{O}(n^2 \log(n))$ .

**Related work.** The standard approach for finding the maximum depth (maximum number of inliers, maximum consensus set etc.) is to use RANSAC [5]. It works by hypothesize and test: Pick a random minimal set (two pairs of correspondences), compute a hypothesis (a translation), and evaluate the hypothesis on the remaining set (the other correspondences), see Algorithm 2. The complexity is  $\mathcal{O}(kn)$ , where  $k$  is the number of iterations. If an exhaustive search over all minimal sets is performed, then the complexity becomes  $\mathcal{O}(n^3)$ . Note that even if an exhaustive search is performed there is no guarantee of finding the maximum depth.

There are two operations that need to be defined for RANSAC. The first one is given two pairs  $(\mathbf{v}_1, \mathbf{v}'_1)$  and  $(\mathbf{v}_2, \mathbf{v}'_2)$ , compute a translation hypothesis  $\mathbf{t}$ . Applying the epipolar constraint twice, one obtains

$$\mathbf{t} = (\mathbf{v}_1 \times \mathbf{v}'_1) \times (\mathbf{v}_2 \times \mathbf{v}'_2). \quad (5)$$

The second operation is to determine whether a correspondence pair  $(\mathbf{v}, \mathbf{v}')$  is feasible or not for a given  $\mathbf{t}$  and residual error  $\epsilon$  in order to count the number of inlier pairs. This is done by first computing the wedge normals  $\mathbf{n}^+$  and  $\mathbf{n}^-$  using the formula in (2) and then checking whether the conditions  $\mathbf{t} \cdot \mathbf{n}^+ \geq 0$  and  $\mathbf{t} \cdot \mathbf{n}^- \geq 0$  are fulfilled or not.

---

**Algorithm 2**  $K$ -iteration RANSAC

---

Input: Correspondence pairs  $(\mathbf{v}_i, \mathbf{v}'_i)$ ,  $i = 1, \dots, n$   
Set maxdepth = 0  
Repeat  $K$  times:  
  For a random subset of two pairs  
    Compute a hypothetical translation  $\mathbf{t}$  using (5)  
    Set depth = #feasible correspondences for  $\mathbf{t}$   
    Update maxdepth

---

## 4. Experiments

In this section we evaluate Algorithm 1 in two different settings. The first problem has no rotation, only translation, between the pairwise images. We compare Algorithm 1 with Algorithm 2 and the well known 5-point algorithm [15]. In the second experiment the direction of the gravitation axis is assumed to be known, leaving us to determine not only the translation, but also one rotation angle. In this experiment we branch and bound over the rotation angle, which we can assume is in the interval  $[-\pi, \pi]$ . In the second experiment we compare Algorithm 1 with a 3-point algorithm [6] and the 5-point algorithm. In both experiments we use real images taken with an Iphone 4. We also add some synthetic experiments where we evaluate the algorithms for different levels of outliers. All algorithms were implemented in C++ using an Intel core i7, 3.4 GHz with 16 GB RAM computer. We did not use more than one of the four cores in any of the experiments, even though all the algorithms are easy to run in parallel. For detecting keypoints we use SIFT [13].

### 4.1. Pure Translation

In this experiment only translated images are considered (no rotation). To make a good evaluation we used 136 image pairs, see Fig. 5. We benchmarked Algorithm 1 against Algorithm 2 and the 5-point algorithm. The number of iterations for Algorithm 2 were set to 500 which gives more than 99 % probability of finding a good solution with up to 90 % outliers. In the 5-point case we ran it the same number of times as Algorithm 1 (10,000 iterations).

To maximize the information gained from the SIFT points, every point in the first image was matched with their best corresponding point in the second image. This resulted in 5,000-10,000 matches for every image pair. We set the maximum angular error  $\epsilon$  so it corresponds to one pixel in the images.

To generate ground truth data we manually selected point matches in every image and used bundle adjustment [8] to calculate the structure and motion. The translation errors with respect to the ground truth for the different algorithms are shown in Fig. 7. The mean angle error for the translation vector can be seen in Table 1. The 5-point algorithm has difficulties since it can force in extra inliers by rotating the space which results in bad translation estimates. In Fig. 6 we compare the number of inliers for the three methods. The 5-point algorithm has 10 % more inliers than Algorithm 1 which in turn has 20 % more inliers than Algorithm 2.

We also compare Algorithm 1 and Algorithm 2 with 500 iterations of LO-RANSAC [1]. The LO-RANSAC algorithm found 2% more inliers than Algorithm 2, which is 17% lower than the number of inliers found by Algorithm 1.

	Alg. 1	Alg. 2	5-pt. alg.
Translation error	4.6°	8.5°	51°

Table 1: The mean translation error measured in degrees between the translations of the different methods and the ground truth translation. Ground truth is calculated with manually selected points and bundle adjustment.

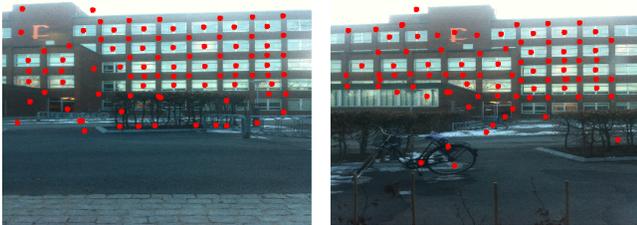


Figure 5: One image pair out of 136, together with some of the SIFT points. There is no rotation between the images, only translation. Note that there are a lot of repeated structures (windows).

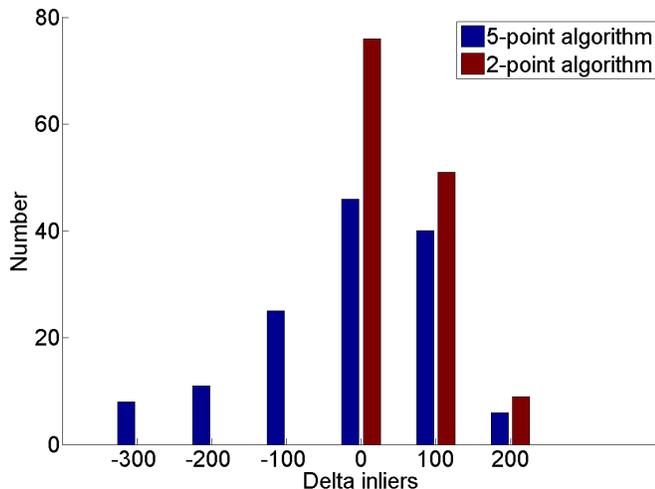


Figure 6: Histogram over the inlier differences obtained from the 136 image pairs. On average there are 7,300 matches between every pair of images. The red bars represent the difference in inliers between Algorithm 1 and Algorithm 2, with 500 iterations. The blue bars represent the difference between Algorithm 1 and the 5-point algorithm, with 10,000 iterations.

#### 4.1.1 Synthetic Experiments

In this synthetic experiment we evaluate how a different number of outlier correspondences affect the number of detected inlier correspondences for Algorithm 2, compared to Algorithm 1. We set  $K \in \{100, 500, \infty\}$  and use 100

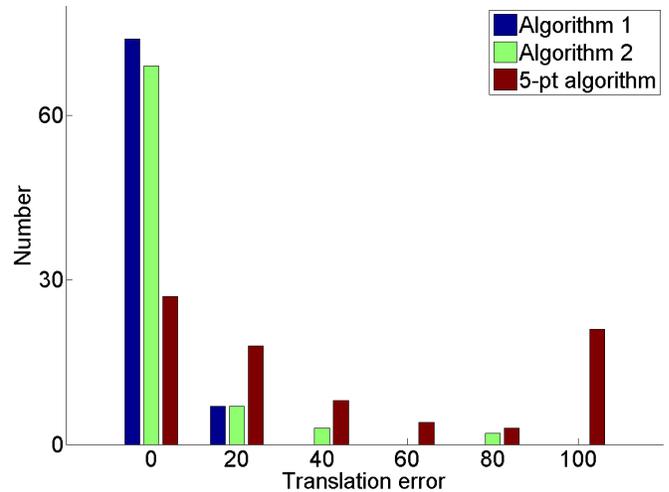


Figure 7: Histogram over the translation errors. The figure compares the estimated translations for Algorithm 1, Algorithm 2 and the 5-point algorithm, with the ground truth translation. The translation error is measured as the angle to the true translation, in degrees.

matches in total. We use the same angular error  $\epsilon$  as in the real experiment above. We also add normally distributed noise with a standard deviation  $\epsilon/3$ . The runtime for Algorithm 2 with  $K = 100$  equals that of Algorithm 1. Here  $K = \infty$  means exhaustive search over all  $\binom{100}{2} = 4950$  possibilities. The results for this experiment are reported in Fig. 8. Note that even though Algorithm 2 checks all the possibilities of two pairs, it does not guarantee the optimal solution as Algorithm 1 does.

#### 4.2. Known Gravitation Direction

In this second experiment we search over the rotation angles as well. This can be done optimally by using Algorithm 1 with branch and bound, similar to the method in [7]. The accelerometer in smart phones registers the gravitation direction, and this can be extracted as sensor data. The gravitation direction for each image can then be rotated so that the images only differ by a translation and a rotation around a known axis. In this experiment we compare the branch and bound version of Algorithm 1 with the 3-point algorithm and the 5-point algorithm. We use 10 images (45 image pairs) for which the ground truth was obtained by manually selecting corresponding points and performing bundle adjustment. The camera rotations were constrained to have the known rotation axes. We use SIFT matching with ratio 0.8, to get 350 correspondences on average for the image pairs. The 3-point algorithm was used with 5,000 RANSAC iterations and the 5-point algorithm with 10,000 iterations. Since we had fewer corresponding points we used three times as large error threshold (corresponding to 3 pix-

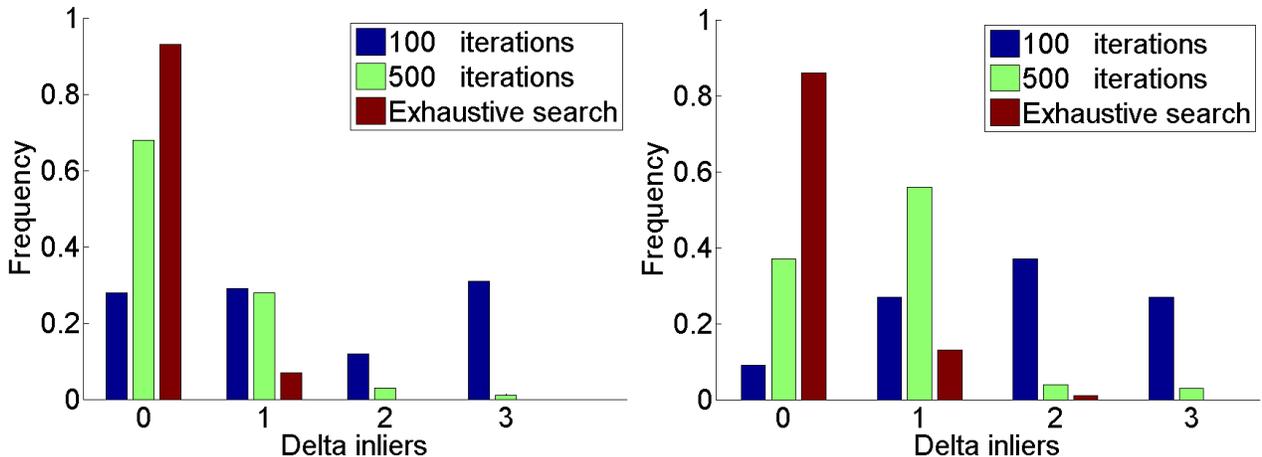


Figure 8: Difference in inliers: Algorithm 1 versus RANSAC with  $K \in \{100, 500, \infty\}$  iterations of Algorithm 2. In this synthetic experiment 100 points were used. Note that 100 iterations take the same time as Algorithm 1 and that  $K = \infty$  corresponds to exhaustive search in a deterministic manner. The optimal solutions for the left and right diagrams have 90 and 95 outliers respectively. The last staple corresponds to differences  $\geq 3$ .

	Branch and bound Alg. 1	3-pt. alg.	5-pt. alg.
Inliers	234	231	242

Table 2: The number of inliers on average for the 45 image pairs, out of an average of about 350 correspondences. The inlier threshold was set to 3 pixels.

Method	Angle	Axis	Translation
Algorithm 1	1.9°	6.5°	9.3°
3-point algorithm	2.0°	6.5°	9.1°
5-point algorithm	2.4°	9.4°	9.4°

Table 3: The mean errors for the angle, axis and translation vector, between Algorithm 1, the 3-point algorithm and the 5-point algorithm. The errors are measured as the angle to the ground truth. In this experiment the gravitational directions for the images are known, but not the angle nor the translation between the images.

els).

The mean errors for the different methods can be seen in Table 3. Note that the 3-point algorithm and the branch and bound version of Algorithm 1 shows very similar results. This is expected since there are a lot of inliers in this experiment. Note also that the rotation axis for the 5 point algorithm is bad. This is probably due to the fact that it has more degrees of freedom, see Table 2. The RANSAC algorithms took about 2-3s while the branch and bound version took about 10s.

#### 4.2.1 Synthetic Experiment

This setup is similar to the previous synthetic experiment with pure translation. Here we randomized the angle and the translation for 100 image points. We let the number of outliers vary, 90 % and 95 % and iterated the 3-point algorithm 500, 5,000 and 20,000 times. We compare this to the branch and bound version of Algorithm 1. Since branch and bound makes Algorithm 1 much slower (about 1.5s), 20,000 iterations of the 3-point algorithm take the same time. The results are reported in Fig. 9. The reason why we do not compare with the 5-point algorithm is that it does not provide in a good solution with this amount of outliers.

## 5. Conclusions

We have shown that the two-view epipolar geometry of a translating camera can be robustly computed in an efficient manner by just using simple arithmetic operations in low-order polynomial time. Unlike random sampling methods, the result is provably optimal and not affected by large amounts of outliers. In fact, for difficult scenarios with large rates of outliers, the results are not only more accurate and reliable, but obtained in less time.

In addition, the proposed method can be embedded into rotation space search framework, similar to [7], making it a robust and reliable method for computing the full motion of the camera. From a computational point of view, it is particularly efficient for small rotations or restricted motions.

**Acknowledgments.** We gratefully acknowledge funding from the Swedish Foundation for Strategic Research (Future Research Leaders) and the Swedish Research Coun-

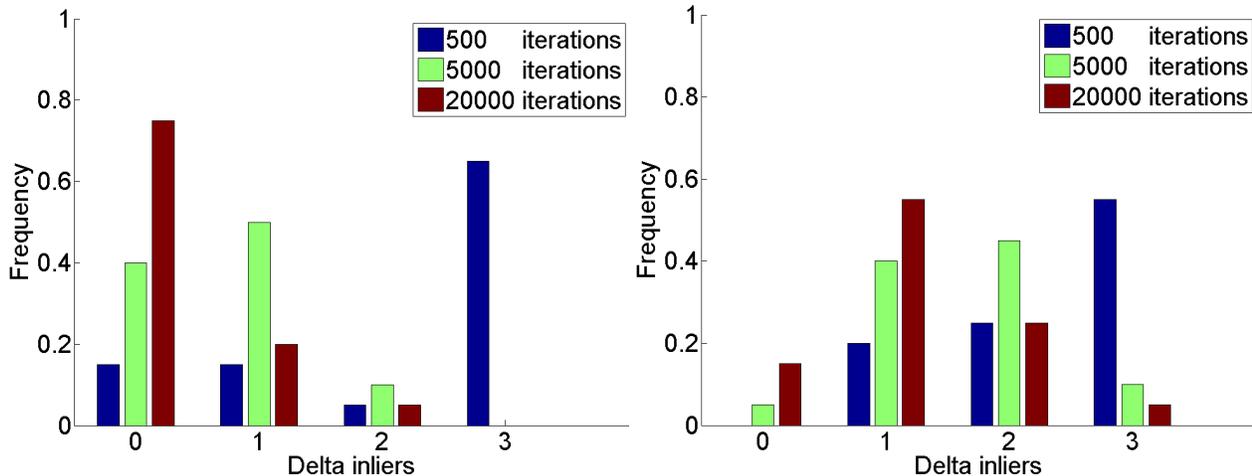


Figure 9: Difference in inliers: Branch and bound with Algorithm 1 versus RANSAC with  $K \in \{500, 5000, 20000\}$  iterations of the 3-point algorithm. In this synthetic experiment, in total 100 points were used. The optimal solutions for the left and right diagrams have 90 and 95 outliers respectively. The last staple correspond to differences  $\geq 3$ .

cil (grant no. 2012-4215).

## References

- [1] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Pattern Recognition*, pages 236–243. Springer, 2003. 4
- [2] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, third edition, 2008. 3
- [3] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *European Conf. Computer Vision*, Florence, Italy, 2012. 1
- [4] O. Enqvist and F. Kahl. Two view geometry estimation with outliers. In *British Machine Vision Conf.*, London, UK, 2009. 1
- [5] 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. 1, 4
- [6] F. Fraundorfer, P. Tanskanen, and M. Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *European Conf. Computer Vision*, Crete, Greece, 2010. 1, 4
- [7] R. Hartley and F. Kahl. Global optimization through rotation space search. *Int. Journal Computer Vision*, 82(1):64–79, 2009. 1, 2, 5, 6
- [8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. Second Edition. 1, 4
- [9] F. Kahl and R. Hartley. Multiple view geometry under the  $L_\infty$ -norm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008. 1, 2
- [10] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1834–1847, 2007. 1
- [11] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Int. Conf. Computer Vision*, Kyoto, Japan, 2009. 1
- [12] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981. 1
- [13] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal Computer Vision*, 60(2):91–110, 2004. 4
- [14] O. Naroditsky, X. Zhou, J. Gallier, S. Roumeliotis, and K. Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(4):812–824, 2012. 1
- [15] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. 1, 4
- [16] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *Conf. Computer Vision and Pattern Recognition*, San Francisco, USA, 2010. 1
- [17] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery using a reference plane. *Int. Journal Computer Vision*, 49(2/3):117–141, 2002. 1
- [18] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter. An adversarial optimization approach to efficient outlier removal. In *Int. Conf. Computer Vision*, Barcelona, Spain, 2011. 1