Outlier Rejection for Absolute Pose Estimation with Known Orientation

Viktor Larsson[1]
viktorl@maths.lth.se
Johan Fredriksson[1]
johanf@maths.lth.se
Carl Toft[2]
carl.toft@chalmers.se
Fredrik Kahl[1,2]
fredrik@maths.lth.se

[1] Centre for Mathematical Sciences,
Lund University,
Sweden

[2] Department of Signals and Systems,
Chalmers University of Technology,
Sweden

Abstract

Estimating the pose of a camera is a core problem in many geometric vision applications. While there has been much progress in the last two decades, the main difficulty is still dealing with data contaminated by outliers. For many scenes, e.g. with poor lightning conditions or repetitive textures, it is common that most of the correspondences are outliers. For real applications it is therefore essential to have robust estimation methods.

In this paper we present an outlier rejection method for absolute pose estimation. We focus on the special case when the orientation of the camera is known. The problem is solved by projecting to a lower dimensional subspace where we are able to efficiently compute upper bounds on the maximum number of inliers. The method guarantees that only correspondences which cannot belong to an optimal pose are removed. In a number of challenging experiments we evaluate our method on both real and synthetic data and show improved performance compared to competing methods.

1 Introduction

Camera pose estimation is a classic problem in computer vision and it is used as a basic building block in many 3D reconstruction pipelines. For instance, in order to add a new image to an existing 3D reconstruction, one can first match image points to the 3D-model, and then compute the camera's pose parameters using the available 2D-3D correspondences. The problem is often referred to as the *absolute pose problem* or *resectioning*. Inevitably, some of the matches will be incorrect, and therefore robust estimation techniques such as RANSAC with a minimal solver [6] are required. For calibrated cameras the minimal case is three point correspondences, see [7, 12]. In [13] Kukelova *et al*. present a minimal solver using only two points with the additional assumption that the rotation axis is known.

There exist several methods for the absolute pose problem with given point correspondences. These methods are often called *Perspective-n-Points* (P*n*P) methods [5, 14, 21, 22]. Most of these methods assume that outliers have been removed in a pre-processing step and consider only the pose estimation problem. A notable exception is the method from Ferraz

Pages 45.1-45.12
DOI: https://dx.doi.org/10.5244/C.30.45

et al. [5] which extends the method from [15] by adding an outlier rejection scheme. The PnP method in [3] uses a branch-and-bound approach to compute the maximum number of inliers in an optimal manner, but at the price of worst-case exponential running time.

In this work we consider the special case where the camera orientation is known. This situation is common in robotics, for example, where the orientation often can be estimated with other sensors, or using some other application specific knowledge. In the experimental evaluation we show one such example where we perform metric localization for a car driving through a tunnel based on a single image. The tunnel has extreme amounts of repetitive textures making it difficult for standard approaches. At the same time, there is a strong prior knowledge on the orientation of the car. The main contribution in this paper is an outlier rejection scheme for this setting. Our method can handle difficult cases with high outlier ratios ($> 99\%$) and is guaranteed to only remove correspondences which cannot be an inlier to the optimal camera position. The idea is to reduce the problem to a one dimensional search, which can be efficiently solved in low order polynomial time. We experimentally compare our approach with the methods in [5, 12]. To our knowledge, there is no competing method that is able to perform outlier rejection for absolute pose with known orientation based on geometric reprojection errors in time $\mathcal{O}(n^2 \log n)$ where n is the number of correspondences.

The idea of outlier rejection was first developed by Svärm *et al.* [19] for city-scale localization. In the paper they present an outlier rejection scheme based on the assumptions that the gravity axis is known in the camera frame and that a plane which contains the focal point is approximately known. The outlier rejection is then performed by projecting the reprojection cones to the known plane and computing bounds in this lower dimensional space. Another work which is conceptually similar to ours is by Wilson and Snavely [20], where they present a method for the different problem of robust relative translation averaging. In the work they consider the case when the orientations are known and the goal is to estimate global camera positions which are consistent with as many pairwise relative estimates as possible. To solve this they employ a similar strategy where they perform outlier rejection by solving one dimensional sub-problems.

2 Background

In this work we consider the problem of estimating the camera pose given 2D-3D point correspondences. We assume that the internal camera calibration is known. For our camera model we use spherical cameras where the projections are formed by scaling to unit length in the camera coordinate system, that is, for a camera with pose parameters $(R, \mathbf{t})$, a point $\mathbf{X}$ is projected to

$$\frac{R(\mathbf{X}-\mathbf{t})}{\|R(\mathbf{X}-\mathbf{t})\|}. \tag{1}$$

For calibrated cameras the most natural error metric is the angular reprojection error. In this metric we say that a 2D-3D correspondence $(\mathbf{x}, \mathbf{X})$ is an inlier to a camera $(R, \mathbf{t})$ if the angle between the reprojection and the image point is less than some fixed threshold ε, i.e.

$$\angle\left(\mathbf{x},\ \frac{R(\mathbf{X}-\mathbf{t})}{\|R(\mathbf{X}-\mathbf{t})\|}\right) \leq \varepsilon \quad \Leftrightarrow \quad \langle \mathbf{x},\ R(\mathbf{X}-\mathbf{t})\rangle \geq \cos(\varepsilon)\,\|R(\mathbf{X}-\mathbf{t})\|, \tag{2}$$

where $\angle$ denotes the angle. If the orientation of the camera is known we can without loss of generality assume, $R = I$, by rotating the image points. Each 2D-3D correspondence then constrains the translation $\mathbf{t}$ to lie in a cone in $\mathbb{R}^3$,

$$\mathcal{K} = \left\{ \mathbf{t} \in \mathbb{R}^3 \;\middle|\; \|\mathbf{X} - \mathbf{t}\| \leq \frac{1}{\cos(\varepsilon)} \langle \mathbf{x},\, \mathbf{X} - \mathbf{t} \rangle \right\}. \tag{3}$$

In the presence of outliers it will not be possible to satisfy all reprojection constraints. Instead we search for a translation which is contained in as many cones as possible. Let $\mathcal{K}_i$ correspond to the cone formed using the ith 2D-3D correspondence. The problem is then equivalent to

$$\max_{\mathcal{I}} \quad |\mathcal{I}| \quad \text{s.t.} \quad \bigcap_{i \in \mathcal{I}} \mathcal{K}_i \neq \emptyset, \tag{4}$$

where $\mathcal{I} \subset \{0, 1, 2, \ldots, N\}$ is the index set for the unknown inlier correspondences. If $\mathcal{I}^\star$ is the optimal inlier set in (4) then any $\mathbf{t}^\star \in \bigcap_{i \in \mathcal{I}^\star} \mathcal{K}_i$ is optimal to the original problem. Note that here optimal is in the sense that it maximizes the number of inlier correspondences for some given threshold ε (and not the total reprojection error). The main difficulty of the problem is to identify the correct 2D-3D correspondences and if necessary the solution can be further refined using standard bundle adjustment methods [10, 11].

3 Outlier Rejection

In this section we present our method for outlier rejection for the absolute pose problem. We assume that the orientation is known or can be estimated using other methods.

For a point correspondence $(\mathbf{x}_0, \mathbf{X}_0)$ we want to decide if it is possible for this correspondence to be an inlier to an optimal translation. If the result is *negative* we can safely remove the correspondence from further consideration. In the notation of (4) this corresponds to determining if the cone $\mathcal{K}_0$ can be part of the optimal solution. If we assume that this is the case then we can reformulate the problem as

$$\max_{\mathcal{I}} \quad |\mathcal{I}| \quad \text{s.t.} \quad \bigcap_{i \in \mathcal{I}} (\mathcal{K}_0 \cap \mathcal{K}_i) \neq \emptyset. \tag{5}$$

The intersection of multiple cones can be very complex. To simplify the problem we orthogonally project each cone intersection $\mathcal{K}_0 \cap \mathcal{K}_i$ to the center line of the cone $\mathcal{K}_0$, see Figure 1. Furthermore since each non-empty intersection $\mathcal{K}_0 \cap \mathcal{K}_i$ is a closed convex set, the projections will form closed intervals $[a_i, b_i] \subset \mathbb{R}$. This projection reduces the problem to one dimension.

Once all of the intervals have been found we compute the maximum number of overlapping intervals. This problem can be solved by simply sorting the interval boundaries and going through them. The maximum number of overlapping intervals will be an upper bound

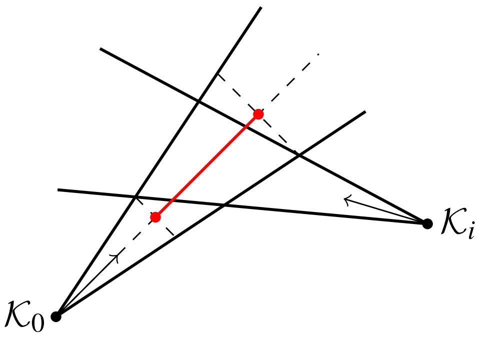

Figure 1: The intersection between the cones $\mathcal{K}_0$ and $\mathcal{K}_i$ forms a closed convex set. Orthogonally projecting this set to the center line in $\mathcal{K}_0$ yields a closed interval.

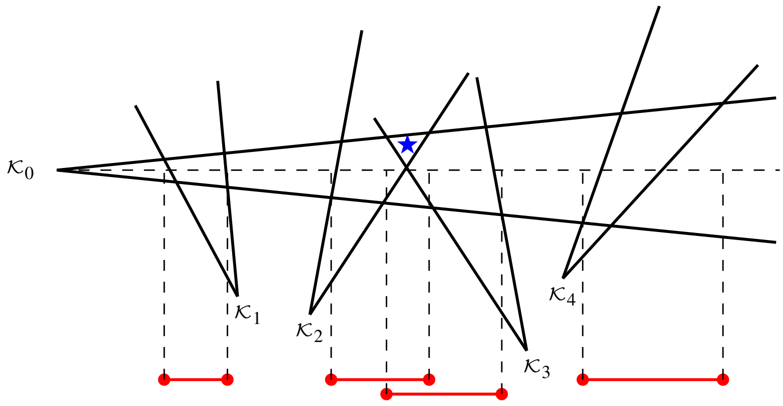

Figure 2: Each intersection $\mathcal{K}_0 \cap \mathcal{K}_i$ is projected to the center line of $\mathcal{K}_0$. These projections form intervals and by counting the number of overlapping intervals we find an upper bound for the number of cones which can simultaneously intersect $\mathcal{K}_0$. In the image the optimal translation is indicated by a blue star.

for the solution in (5). See Figure 2. The procedure is summarized in Algorithm 1.

```
for each cone $\mathcal{K}_i$ do
    $\ell_i \leftarrow$ center line in $\mathcal{K}_i$
    for each cone $\mathcal{K}_j$ do
        $[a_j, b_j] \leftarrow$ project $(\mathcal{K}_i \cap \mathcal{K}_j)$ onto $\ell_i$
    end
    Sort the boundary points $\{a_j\}$ and $\{b_j\}$
    Loop through the boundaries keeping track of the number of overlaps
    if the number of overlaps is less than the best known solution then
        Mark the cone $\mathcal{K}_i$ as an outlier
    end
end
```

Algorithm 1: Outlier rejection

4 Finding the Intersection Bounds

To find the interval corresponding to the projection of $\mathcal{K}_0 \cap \mathcal{K}_i$ we want to solve the problem

$$\begin{aligned} \min/\max \quad & \langle \mathbf{x}_0, \mathbf{t} \rangle && (6) \\ \text{s.t.} \quad & \|\mathbf{X}_0 - \mathbf{t}\| \leq \alpha \langle \mathbf{x}_0, \mathbf{X}_0 - \mathbf{t} \rangle && (7) \\ & \|\mathbf{X}_i - \mathbf{t}\| \leq \alpha \langle \mathbf{x}_i, \mathbf{X}_i - \mathbf{t} \rangle, && (8) \end{aligned}$$

where $\alpha = 1/\cos(\varepsilon)$ and $\mathbf{x}_0$ is the unit direction of the center line. This is a conic optimization problem and can be solved by standard solvers such as MOSEK [16] or SeDuMi [18]. Unfortunately since we need to solve many instances to compute the bounds for a single correspondence, these methods are too computationally expensive to be of practical use. Therefore in the next section we show how to quickly compute bounds by replacing cones with outer planar approximations.

4.1 Planar Approximations

To simplify the problem of computing the interval bounds we approximate the cones with polyhedral sets. For each cone $\mathcal{K}_i$ we find a set of tangent planes such that

$$\mathcal{K}_i \subset \{\, \mathbf{t} \mid \langle \mathbf{a}_k, \mathbf{t} \rangle \leq b_k,\ k = 1, 2, \ldots, n_p \,\}. \quad (9)$$

The plane normals $\mathbf{a}_k$ are chosen such that they are evenly distributed around the cone. This is illustrated in Figure 3. In Section 6.1 we experimentally evaluate how the number of planes affect the solution. Of course as the number of planes tend to infinity the approximation error vanishes.

When we compute the projections we only need to approximate the cone on which we perform outlier rejection on (i.e. $\mathcal{K}_0$). The problem in (6) then becomes

$$\begin{aligned} \min/\max \quad & \langle \mathbf{x}_0, \mathbf{t} \rangle && (10) \\ \text{s.t.} \quad & \langle \mathbf{a}_k, \mathbf{t} \rangle \leq b_k, \quad k = 1, 2, \ldots, n_p && (11) \\ & \|\mathbf{X}_i - \mathbf{t}\| \leq \alpha \langle \mathbf{x}_i, \mathbf{X}_i - \mathbf{t} \rangle. && (12) \end{aligned}$$

The approximation allows us to solve the problem by enumerating the possible Karush-Kuhn-Tucker (KKT) points [2]. Each KKT point can be solved for in closed form. Additionally we need to check if the problem is unbounded, which will result in an unbounded interval $[a_i, \infty)$. More details and derivations can be found in the supplementary material.

5 Pose Estimation Pipeline

Our full pipeline for pose estimation is first running the outlier rejection, followed by a few iterations of RANSAC on the remaining correspondences. In each iteration of Algorithm 1 we also compute a translation candidate by taking a point on the center line ℓ_i which lies in the maximal interval. This simple heuristic gives a good lower bound (i.e. an actual solution) which allows us to more quickly discard outliers and hence achieve improved runtimes.

When the orientation is fixed there remain three degrees of freedom in the camera pose. This means that with one point correspondence the problem is under-constrained and with two it is over-constrained. This can be resolved by ignoring one coordinate for one of the

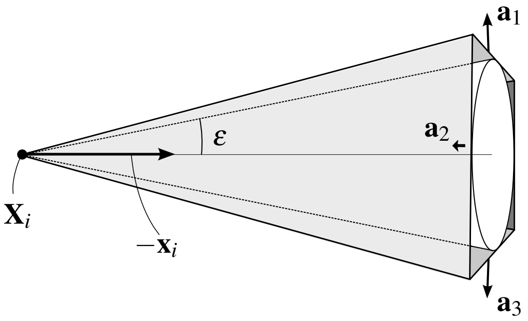

Figure 3: A cone approximated by a set of planes ($n_p = 4$ in this case). The apex is located at $\mathbf{X}_i$ and its axis is directed along $-\mathbf{x}_i$. The vectors $\mathbf{a}_k$ correspond to the plane normals.

points and solving for the translation using only the three remaining coordinates. In this case the problem reduces to a simple linear system.

In our RANSAC we sample two points and construct four translation estimates by discarding each of the four coordinates. We then keep the translation which gives the smallest reprojection error for the discarded coordinate. In the experiments we denote this 1.5 point RANSAC.

6 Experimental Evaluation on Synthetic Data

6.1 Evaluation of the Planar Approximation

In this section we evaluate how the number of planes affect the approximation error. We generated random pairs of cones and computed the intersection intervals, that is, solving min/max problem (10) for the planar approximation using a varying number of planes. For each pair we also computed the intersection interval without any approximation using CVX [8]. Figure 4 (left) shows the average relative interval size for different number of planes over 1000 instances. We can see that using four planes only gives an average approximation error of about 10%.

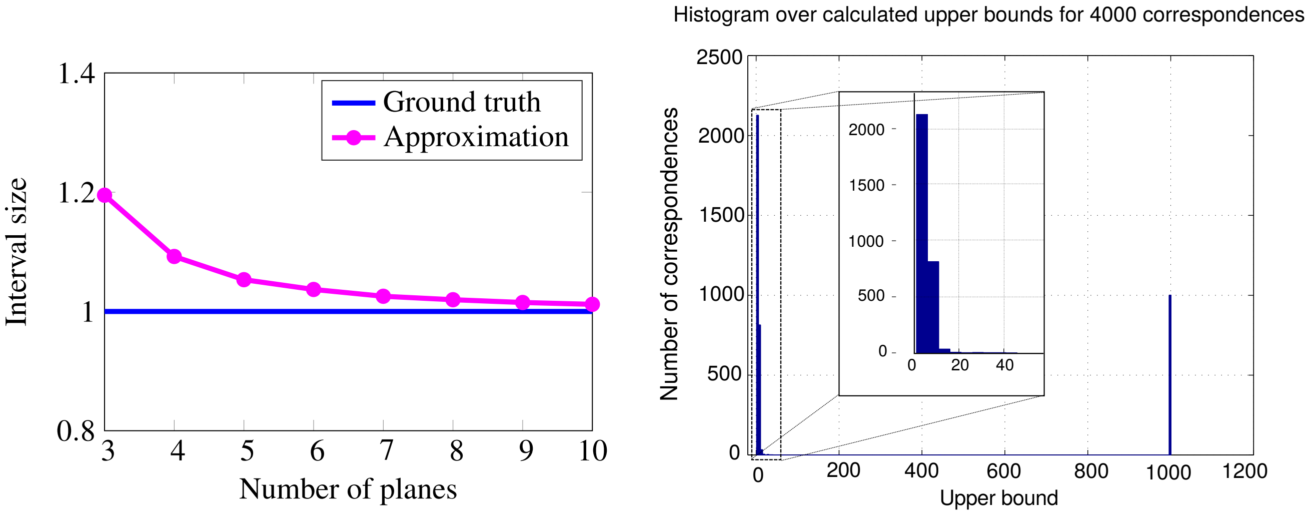

Figure 4: *Left:* Evaluation of the interval size for the planar approximation. The sizes are relative to the ground truth interval size. *Right:* Histogram over upper bounds obtained during outlier rejection performed on 4000 correspondences with 75% outliers.

6.2 Evaluation of the Upper Bounds

In this section we evaluate the upper bounds computed by the outlier rejection and how the planar approximation described in Section 4.1 affects them.

First, upper bounds were calculated using the planar approximation by generating 4000 synthetic 2D-3D correspondences and performing the outlier rejection scheme described in Section 3. Points were generated from a uniform distribution in the box $[-3,3] \times [-3,3] \times [7,13]$, and a spherical camera was similarly generated in the cube $[-1,1] \times [-1,1] \times [-1,1]$. After projecting the 3D points into the camera, 75% of the 3D points were re-sampled in order to create outlier correspondences. The angle of the error cones was set to $\varepsilon = 0.1^{\circ}$. Figure 4 (right) shows a histogram over the upper bounds found using the planar approximation.

Note that the majority of the correspondences have an upper bound less than 20, and that there is a peak at 1000. The peak corresponds to the 25% inliers, and all correspondences with an upper bound less than 20 can be safely classified as outliers and permanently discarded.

Finally we compared the upper bounds calculated using the planar approximation with the upper bounds from the exact solution found using CVX [8]. For this, 100 correspondences were generated as described above, and upper bounds were calculated using both methods. This was repeated 10 times. The upper bounds were identical in both methods for over 90% of the correspondences, the rest differing only by one from the exact solution. This suggests that the planar approximation approach can be safely used in place of the much slower exact solution. The runtime for the CVX based code was about 20-30 minutes per instance, while the outlier rejection using the planar approximations (implemented in C++) ran in a couple of milliseconds.

6.3 Comparison to Other Methods

In this section we compare the performance of the outlier rejection to other methods for robust absolute pose estimation. We compare with RANSAC using both the 1.5p solver (known orientation) and the 3p solver (full pose) from [12]. Furthermore we compare with the recent method from Ferraz *et al.* [5] which solves the P*n*P problem using an algebraic outlier rejection scheme.

Similarly to the experimental setup in [5], we generate synthetic cameras with image size 640×480, focal length 800 and principal point in the center of the image. For each instance we sampled 1000 3D points uniformly from the box $[-2,2] \times [-2,2] \times [4,8]$ in the camera's coordinate system. After projecting the points we resampled a subset of the 3D points, thus creating some outlier correspondences. To the projections we added Gaussian noise with 2 pixel standard deviation.

Using our method we performed outlier rejection followed by 10 iterations of 1.5p RANSAC on the remaining points. For the other two RANSAC methods we ran 100 iterations, which makes the runtime approximately the same for our method and 1.5p RANSAC (about 30-40ms per instance). Figure 5 shows the results from the experiment. We present our results both with and without the additional 1.5p RANSAC iterations. The number of inliers is measured relative to the number of inliers for the ground truth pose. Note that due to the added noise the ground truth pose is not necessarily the optimal pose. Figure 5 also shows the relative translation error $||\mathbf{t} - \mathbf{t}_{GT}|| \, / \, ||\mathbf{t}||$.

The experiment shows that our method is able to produce good estimates for problems with extremely large outlier ratios. Furthermore we can see that the initial estimate produced by the outlier rejection is very good. On average we were able to remove 96.7% of the outliers.

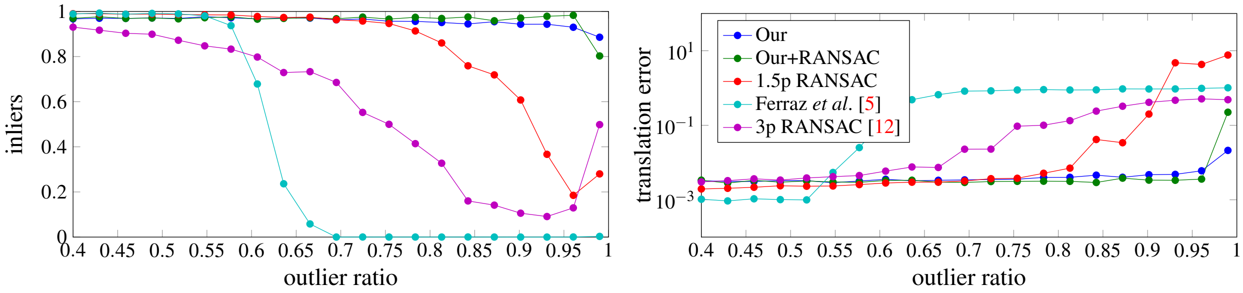

Figure 5: Comparison with other methods over 100 random instances. *Left:* The number of inliers relative to the number of inliers for the ground truth pose. The peak at 99% inliers for the RANSAC methods is due to the optimal inlier set approaching the size of the minimal sets. *Right:* The relative translation error ($\|\mathbf{t}-\mathbf{t}_{GT}\| \,/\, \|\mathbf{t}_{GT}\|$).

7 Experimental Evaluation on Real Data

In this section, we evaluate our framework on real data. Our application of interest is to develop an off-line verification system which is capable of determining the accuracy of a self-driving car's internal odometry based on dead-reckoning. This is typically done with differential GPS which has an accuracy of up to a few centimetres. However, for the scenario we consider (a tunnel), the GPS signal is not accessible and there is a need for an independent way of checking the accuracy. This application will serve as a test bed for our framework.

The first step is to construct a 3D model of the tunnel. We have used a publicly available structure from motion package ([4]) based on SIFT features. In total, 600 images covering approximately 300 m taken from a car were used. The 3D model consists of 904,038 points each equipped with a SIFT descriptor[1]. See *supplementary material* for a video of the 3D model. As a second step, we have acquired two independent image sequences, consisting of 544 images in total, see Figure 6 for some examples. All images were taken with a GoPro Hero4 Black at 120 fps with resolution 1920 × 1080 pixels. Further, the exposure was set to ISO 6400 to handle the low-light conditions. Internal camera calibration was obtained using MATLAB Calibration Toolbox [1].

For the test images, there is no ground truth available. In order to create a so-called *gold standard*, that is, the best possible benchmark test under current conditions, we proceed as follows. Assume for a moment that the absolute pose of the first image in the sequence with respect to the 3D model is available. As the speed of the car is (approximately) known as well as its direction, we can predict the position at the next time frame. At this position, we can project the 3D model points and compare to the extracted SIFT features in the image. If the SIFT features match according to the standard Lowe criterion and are within 10 pixels of the image point, we consider it a plausible 2D-3D correspondence. The camera pose is then refined using bundle adjustment [10]. This process is iterated throughout the whole sequence. The first image, taken outside the tunnel, contains an abundance of matches. Hence, we can reliably compute the camera pose with standard 3-point RANSAC followed by bundle adjustment. We have visually checked that the obtained correspondences are correct.

[1] For an actual verification system, one would also need to verify the 3D model accuracy.

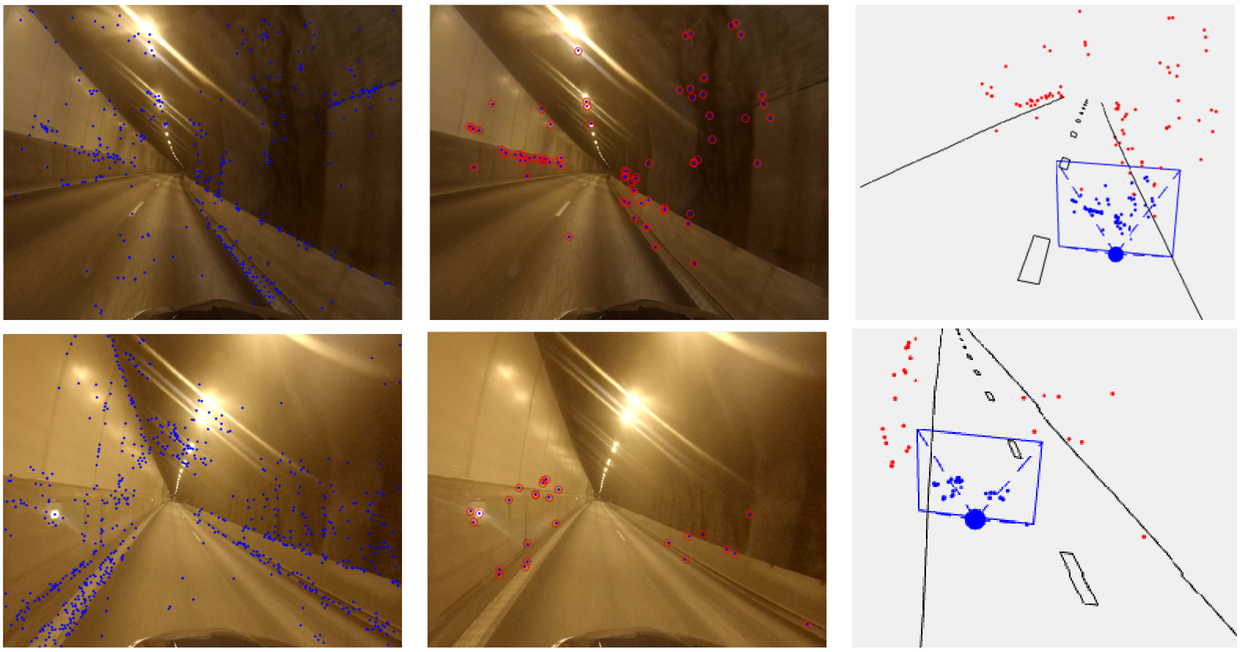

Figure 6: Two pose estimation examples from the tunnel experiment. *Left column:* Input images with SIFT features (blue points). *Middle column:* Reprojected 3D model (red circles) of inlier correspondences. *Right column:* Camera pose in world coordinate frame.

The complete 3D camera trajectory has also been visually inspected to ensure consistency.

Now to the actual benchmark test. For each image, we extracted SIFT features from the image and matched each descriptor to the 5 closest in the 3D model using FLANN [17]. Note that this produces at least 80% outliers, but it also ensures that we are likely to obtain some correct matches despite repetitive patterns (lane markings, lights etc.). The estimated camera position is considered correct if it lies within 0.10 m of the gold standard position. We compare our approach of outlier removal (Section 5) with 3-point RANSAC [6, 12] with varying levels of iterations. Both methods use the inlier threshold $\varepsilon = 0.2^\circ$ and perform bundle adjustment for refinement on the inlier set. The camera orientation is only approximately known as the tunnel turns. Therefore, for our method, we test 10 different pre-set orientations. Still, our method is several factors faster than 3-point RANSAC with 1000 iterations (the speedup depends on the number of correspondences as well as implementation). See Figure 6 and *supplementary material* for a video of the localization results.

As can be seen in Table 1, our approach compares favourably to RANSAC in terms of localization success rate. The main difference is that our approach is able to cope with the high rates of outliers, which is supported by the fact that RANSAC improves as the number of iterations increases. In the outlier rejection step, similar to the synthetic case, most of the outliers (typically more than 90%) are discarded. By inspection, we have found that the failure cases are mainly due to the lack of discriminative image features. Saturation due to light sources and reflections are also reasons for failure. In most of the failure cases, there are simply not enough of correct correspondences. Another source of errors are the pre-set orientations. Figure 7 shows the error between the ground truth rotation and the closest of the 10 pre-set rotations for the 544 test images.

Method	Outlier rejection	3p RANSAC 1,000 iterations	3p RANSAC 10,000 iterations	3p RANSAC 50,000 iterations
Success rate	96%	44%	74%	88 %

Table 1: Metric localization on the tunnel benchmark set (544 images). Images are considered correct if the estimated camera position is within 0.10 m of the gold standard position.

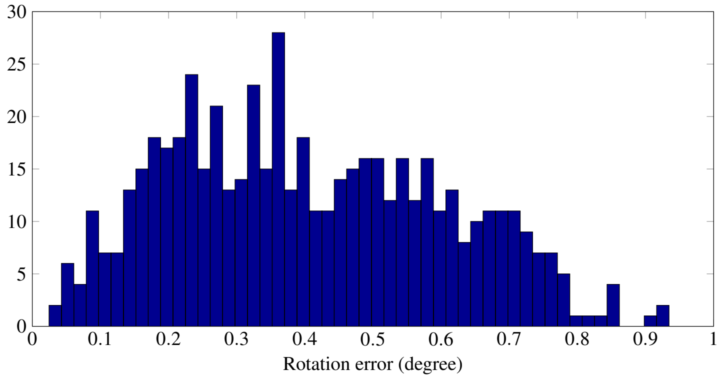

Figure 7: Histogram over the minimum angle between the ground truth rotation and the 10 pre-set rotations used for the 544 test images. The error should be compared to the inlier threshold ε which was set to 0.2 degrees.

8 Conclusions

In this paper we have presented a simple and efficient method for outlier rejection for absolute camera pose estimation in the case of known orientation. The method is guaranteed to remove only true outliers to the optimal solution. In the experiments, both on synthetic and real data, we have shown that the method is able to remove most outliers, which greatly simplifies the problem. To solve the remaining pose estimation we have used RANSAC but this could be substituted for other pose estimation methods. By using polyhedral approximations we achieve significant computational speedups without compromising solution quality.

We have also presented an application in a very challenging setting: Metric localization from a single image in a tunnel. The images are taken in low-light conditions (hence noisy) and contain repeated patterns, reflections, saturations and many times, little texture. Still, we have shown that it is possible to solve the localization problem provided that one can generate sufficiently many correct inlier correspondences. A possible extension of our method is to perform rotation search when the orientation is unknown. For instance, our method can be used in conjunction with the branch-and-bound framework of Hartley and Kahl [9].

9 Acknowledgements

We gratefully acknowledge funding from the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots), the Swedish Research Council (grants no. 2012-4213 and 2012-4215) and the Crafoord Foundation.

References

[1] J.-Y. Bouguet. Camera calibration toolbox for matlab. 2004.

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] O. Enqvist and F. Kahl. Robust optimal pose estimation. In *European Conference on Computer Vision (ECCV)*, Marseille, France, 2008.

[4] O. Enqvist, C. Olsson, and F. Kahl. Non-sequential structure from motion. In *Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras (OMNIVIS)*, Barcelona, Spain, 2011.

[5] L. Ferraz, X. Binefa, and F. Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, USA, 2014.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] X. Gao, X. Hou, J. Tang, and H. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(8):930–943, 2003.

[8] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[9] R. Hartley and F. Kahl. Global optimization through rotation space search. *International Journal of Computer Vision (IJCV)*, 82(1):64–79, 2009.

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

[11] Y. Jeong, D. Nistér, D. Steedly, R. Szeliski, and I. Kweon. Pushing the envelope of modern methods for bundle adjustment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(8):1605–1617, 2012.

[12] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, US, 2011.

[13] Z. Kukelova, M. Bujnak, and T. Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *Asian Conference on Computer Vision (ACCV)*, Pondicherry, India, 2011.

[14] S. Li, C. Xu, and M. Xie. A robust $O(n)$ solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(7):1444–1450, 2012.

[15] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative $O(n)$ solution to the pnp problem. In *International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, 2007.

[16] APS Mosek. The mosek optimization software. *Online at http://www. mosek. com*, 54, 2010.

[17] M. Muja and D.G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisboa, Portugal, 2009.

[18] I. Polik, T. Terlaky, and Y. Zinchenko. Sedumi: a package for conic optimization. In *IMA workshop on Optimization and Control, Univ. Minnesota, Minneapolis*, 2007.

[19] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3d models. In *Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, USA, 2014.

[20] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision (ECCV)*. Zurich, Switzerland, 2014.

[21] Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström, and M. Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *International Conference on Computer Vision (ICCV)*, Sydney, Australia, 2013.

[22] Y. Zheng, S. Sugimoto, and M. Okutomi. Aspnp: An accurate and scalable solution to the perspective-n-point problem. *IEICE TRANSACTIONS on Information and Systems*, 96(7):1525–1535, 2013.