Accurate Localization and Pose Estimation for Large 3D Models

Linus Svärm[1] Olof Enqvist[2]
[1]Centre for Mathematical Sciences
Lund University, Sweden
{linus,magnuso}@maths.lth.se

Magnus Oskarsson[1] Fredrik Kahl[1,2]
[2]Department of Signals and Systems
Chalmers University of Technology, Sweden
{olof.enqvist,fredrik.kahl}@chalmers.se

Abstract

We consider the problem of localizing a novel image in a large 3D model. In principle, this is just an instance of camera pose estimation, but the scale introduces some challenging problems. For one, it makes the correspondence problem very difficult and it is likely that there will be a significant rate of outliers to handle.

In this paper we use recent theoretical as well as technical advances to tackle these problems. Many modern cameras and phones have gravitational sensors that allow us to reduce the search space. Further, there are new techniques to efficiently and reliably deal with extreme rates of outliers. We extend these methods to camera pose estimation by using accurate approximations and fast polynomial solvers. Experimental results are given demonstrating that it is possible to reliably estimate the camera pose despite more than 99% of outlier correspondences.[1]

1. Introduction

A classic problem in computer vision is estimating the orientation and position of a camera, given positions of a number of points in 3D and their projections in the camera image. The so-called *pose estimation problem* has been solved in many contexts and for many camera models, see [10].

Another problem that has attracted increasing attention over the past years is the *localization problem*, i.e. estimating the position (and sometimes the orientation) of a viewer or a camera given image data. A number of approaches have been suggested for solving this problem. Many have adopted an image retrieval approach, where a query image is matched to a database of images using visual features. Sometimes this is combined with a geometric verification step, but in many cases the underlying geometry is largely ignored, see [11, 25, 12].

[1]This work was supported by the Swedish Research Council (grant no. 2012-4215) and the Swedish Foundation for Strategic Research, within the programmes RIT08-0043 and Future Research Leaders.

The approach that we pursue in this paper is viewing the localization problem as a pose estimation problem by matching an image to a large 3D model of the environment. In such an approach one crucial step is the robust matching of image features to features in the 3D model. The ability to handle massive amounts of outliers in the data is absolutely paramount.

For many practical applications, using e.g. vehicle mounted cameras or devices with accelerometers such as smart phones, we can assume that the direction of the gravitational vector is known. This simplifies the problem by reducing the search space and we show that this enables tractable, efficient, robust and accurate algorithms for localization. A key observation is that the problem can be recast as a particular type of registration problem: from points to cones. We use this formulation and present a number of algorithms for performing outlier removal and pose estimation in low-order polynomial time.

The main contributions of the paper are

- Reformulating the pose estimation problem as a registration problem.
- A fast approximate outlier rejection scheme, that enables us to handle large datasets with very large amounts of outliers.
- An optimal algorithm for inlier optimization, that runs in polynomial time.

1.1. Related work

A number of solutions have been proposed for solving the localization problem as a camera pose problem via 2D-to-3D matching, see [19, 5, 18, 23, 24]. The main focus has been to develop sophisticated heuristics for finding reliable matching schemes and avoiding to generate false correspondences. We take a radically different standpoint: Instead we allow the matching scheme to generate a lot of correspondences - correct or incorrect - in order to make sure that we do not miss any good correspondences. The focus of our approach is on the ability to handle large amount of outliers in a reliable and tractable manner.

Many approaches for robust estimation based on the RANSAC framework have been proposed over the years, see e.g. [6]. Although this works well in many cases, one problem with these approaches is that there is no guarantee that they will obtain a reasonable solution even if there exists one. It can also be hard to determine if there is no solution at all. In addition, the number of iterations required to find a solution with high probability tends to make the approach impractical for the rates of outliers that we consider.

Another approach for handling outliers in a robust way is the L_∞-framework, see [13, 14, 26] including recent extensions [22, 28]. Many of these approaches work well for large scale problems, but break down with large rates of outliers.

Solving computer vision problems using IMU or accelerometer data in addition to visual data has been proposed in a number of previous papers. Some use it together with RANSAC, [9, 15], while others use it to bootstrap the filtering process in SLAM type approaches, see [20, 27].

The most similar works to ours include [3, 16, 17, 8] where the aim is to develop algorithms which provably optimizes a robust error norm. In most cases this simply means minimizing the number of outliers but some [2] also consider the truncated L_2 norm. Several of these approaches are based on branch-and-bound which has exponential time complexity. To our knowledge, none of the above approaches are able to solve the pose problem with a provably optimal algorithm based on a robust error criterion that runs in polynomial-time.

2. Problem Formulation

Camera pose estimation is the task of localizing an image in a 3D model. We will make three assumptions. First of all we assume that the query image has known orientation with respect to the ground plane. Typical cases are photos from a camera mounted on a vehicle or photos from a smart phone with accelerometers that measure the gravitational vector when stationary. Secondly, we assume that the direction of gravity is known in the 3D model. This is the case if the images used in the reconstruction where equipped with similar measurements as the query image. Finally, we assume that the ground plane has been roughly located in the 3D model. One way to do this is by considering the height of the cameras used in the reconstruction. The rough location of the ground plane is not necessary for our approach, but it will increase the speed significantly.

Choose a coordinate system such that the camera is at the origin and the z-axis points upwards. Let the 3-vector U denote a 3D-point and let u be a hypothetical correspondence in the image. The relative orientation between the camera and the point is known up to a rotation about the z-axis. In the noise-free case each correspondence should satisfy,

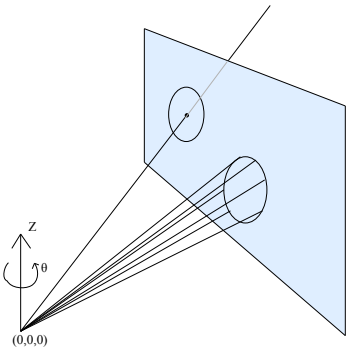

Figure 1. The image plane of the camera is depicted with error ellipses for two example points.

$$\lambda u = SU' = RU + t, \tag{1}$$

with

$$R = \begin{pmatrix} \bar{R} & 0 \\ 0 & 1 \end{pmatrix}, \tag{2}$$

where S is a known 3×3 rotation matrix (encoding the known rotation axis) and $\bar{R}$ is an unknown 2×2 rotation matrix. For ease of notation, and no loss of generality, we will in the following assume that S is equal to the identity matrix.

Since finding accurate correspondences is difficult, we need to solve this problem in a robust way. A common approach is to simply optimize the number of consistent measurement, i.e. inliers. Allthough this formulation leads to a challenging optimization problems, using recent advances in robust estimation it is indeed possible to solve it in polynomial time as function of the number of correspondences.

2.1. A Reformulation

First, we need to define more precisely how to measure an error. In multiple view geometry, measuring reprojection errors is normally the preferred choice, as it accurately models the limited precision of feature detection techniques.

Let the 3D point be rotated and translated to the camera coordinate system. It is easy to show that the set of points in $\mathbb{R}^3$ that yields a reprojection error smaller than a threshold ϵ, forms a cone C in $\mathbb{R}^3$. A 3D point U is an inlier if

$$U' = RU + t \tag{3}$$

lies inside this cone C. Hence we have transformed the original camera pose problem to a registration problem, namely that of registering a number of 3D points U_i to the corresponding cones C_i, see Figure 5.

2.2. Overview of the Approach

In [8] it was shown how the number of outliers can be minimized in polynomial time. The theoretical result is a straightforward consequence of the theory of KKT points. One requirement is that the parameter space should be a differentiable manifold embedded in $\mathbb{R}^m$ with a set of equality constraints. The trick is to introduce a dummy goal function and then construct an algorithm for computing the complete set of KKT points to the resulting optimization problem. For the details we refer to that paper. In order to do this, we need to define a goal function on the parameter space and then construct a set of solvers. The details of how this is done is described in Section 4. The main theorem from [8] shows that one of the solution points generated in this way will be optimal with respect to the number of outliers. In this way we can minimize the number of outliers in $\mathcal{O}(n^5)$ time.

In many cases this is far too slow to be practical. Hence, [2] proposes a simple and fast outlier rejection method to be used as preprocessing step to the optimal estimation. The technique is specialised to the problems of stitching and 2D-2D registration, so we need to generalize it to work in our setting as well. Our method for fast outlier rejection is described in Section 3.

This gives us our complete localization pipeline. We start by matching SIFT features between the image and the model. We then run Algorithm 1 to quickly eliminate a large amount of wrong matches. Finally we run Algorithm 2 to find the best solution. In a number of experiments we show in Section 5 that this approach works for both very large models and for outlier rates up to more than 99%.

2.3. A Note on Errors in Orientation

Modern MEMS accelerometers are incorporated in many of today's hand-held devices such as mobile phones and tablet computers. These accelerometers make it possible to measure the gravitational vector when the device is stationary. The sensitivity of such measurements has increased over the last years, and the typical accuracy is around 1mg which corresponds to an error less than 0.1°. However there is also a zero g level error offset which is typically about 1°. This can in some cases be calibrated away, if the same device is used. If there is a slight motion during the capture of the image, the accelerometer will not correctly measure the gravitational vector, but this can be compensated for, using the gyroscopes which are present in most devices. Typical error values are from the ST Electronics sensor LIS331DLH, see [1] for details.

In our setup these errors can easily be incorporated by increasing the size of the error cones. If we decompose the errors in orientation, in tilt and roll angles, errors in tilt will be most significant. For the roll angle the impact of the error will increase for points farther from the centre of the image.

3. Fast Outlier Rejection

We will now describe our outlier rejection step.

3.1. Connection to Registration

Assume for a moment that the height of the camera, relative to the 3D reconstruction, is known. Then we get a special form of the registration problem from Section 2.1. In this case, the transformation can be written

$$R = \begin{pmatrix} \bar{R} & 0 \\ 0 & 1 \end{pmatrix}, \quad t = \begin{pmatrix} t_x \\ t_y \\ 0 \end{pmatrix}. \tag{4}$$

As this transformation will never change the height of a given 3D point

$$U = \begin{pmatrix} v \\ h \end{pmatrix}, \tag{5}$$

it will always intersect the corresponding cone at $z = h$. Hence we can restrict the cone constraint to this plane and rather than a cone we get a conic section. The 3D point, U, is an inlier if

$$U' = RU + t \tag{6}$$

lies inside this conic section. Note that the third coordinate can be dropped. Hence we have a problem of 2D-2D-registration of points to conic sections.

As we discussed in Section 2, the height will not be known exactly but only restricted to an interval,

$$h_l \leq h \leq h_u. \tag{7}$$

In that case, the registration problem described in Section 2.1 is turned into registering points to cones that are cut by two planes; see Figure 2. As we are looking for lower bounds on the number of outliers, we are free to consider a relaxation of this problem. Hence we throw away the height information by projecting the cut-off cones onto the ground plane. The projected shape is the convex hull of the conic section at height h_l and the conic section at height h_u, but we will get a good approximation using quadrilaterals, see Figure 2. When the conic sections are not ellipses, but rather parabolas, or one-sided hyperbolas, we extend the quadrilaterals to infinity.

Again we have transformed the original camera pose problem to a registration problem, in this case that of registering a number of 2D points v_i to the corresponding quadrilaterals Q_i.

3.2. Rejection Using Minkowski Sums

The basis of our outlier rejection scheme is a bounding function of the following kind: *If correspondence i is an inlier, then there can be no more than B_i inliers.* We will soon see how to achieve such a bound and also how to produce a

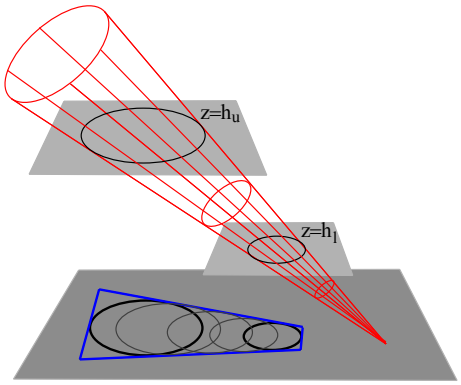

Figure 2. Cutting a cone with a number of planes between two heights $z = h_l$ and $z = h_u$ and projecting them onto the ground plane results in a shape (black) that can be approximated with a quadrilateral (blue).

lower bound, L, on the number of inliers. If $B_i < L$, then correspondence i can safely be removed from the problem.

First recall the definition of Minkowski addition from geometry. The Minkowski difference is defined in an analogue way.

Definition 1. *The Minkowski sum of two sets of position vectors A and B is the set*

$$\{a + b : a \in A, b \in B\}. \tag{8}$$

In Figure 3 an example of the Minkowski difference of two quadrilaterals is shown.

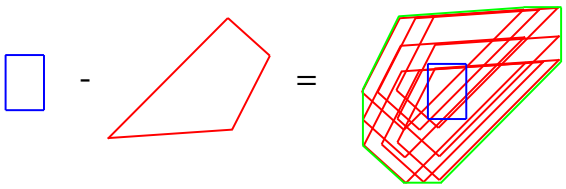

Figure 3. Illustration of the Minkowski difference of two quadrilaterals. The green outline shows the difference of the red and blue quadrilaterals.

Recall from the last section that the problem at hand is registration of points v_i to quadrilaterals Q_i. For technical reasons we pick a point q_i from each Q_i, e.g. the centre of mass. Let $\bar{Q}_i = \{x : x + q_i \in Q_i\}$, i.e. the quadrilateral Q_i translated to the origin.

Theorem 1. *Assume there exist planar R, t such that*

$$Rv_i + t \in Q_i \tag{9}$$

where v_i are 2D points and Q_i quadrilaterals with centres at q_i, for $i = 0, \ldots, k$. Then there exists R, t' such that $Rv_0 + t' = q_0$ (with zero error) and

$$Rv_i + t - q_i \in (\bar{Q}_i - \bar{Q}_0) \tag{10}$$

for all the other i's, where $\bar{Q}_i - \bar{Q}_0$ refers to the Minkowski difference of the two quadrilaterals.

Proof. Choose $t' = -Rv_0 + q_0$. Then

$$\begin{aligned} Rv_i + t' - q_i = Rv_i + t' - q_i - t + t = \\ = (Rv_i + t - q_i) - (Rv_0 + t - q_0). \end{aligned} \tag{11}$$

As $(Rv_i + t) \in Q_i$, then by definition $(Rv_i + t - q_i) \in \bar{Q}_i$. A similar argument shows that $(Rv_0 + t - q_0) \in \bar{Q}_0$ and hence the difference in (11) lies in $\bar{Q}_i - \bar{Q}_0$. □

This theorem allows us to set the error of one point to zero, by expanding the uncertainty regions around the other points. The number of inliers with respect to the Minkowski differences, given that point 0 is error-free gives us a bound of the desired type: *If correspondence 0 is an inlier, then there can be no more than B_0 inliers.*

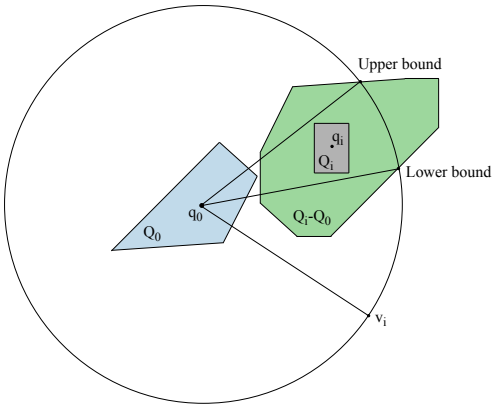

Figure 4. Propagating the errors from point v_0 to v_i can be done using the Minkowski difference of the error. The bounds on the rotation for a point v_i are also shown.

See Figure 4 for a depiction of the setup for one point v_i. Constraining the first point to be error-free fixates the translation. Hence we have a set of points and a collection of sets $M_i = \bar{Q}_i - \bar{Q}_0$ and we wish to find a rotation that maximizes the number of points enclosed in their corresponding sets. As should be clear from the figure, each point-to-set correspondence will be consistent with all θ's in intervals I_i. By sorting all interval boundaries and going through the sorted list, we can find an optimal choice of θ. The computational cost of this is $\mathcal{O}(n \log n)$. For the angle with maximally many consistent points, we also reproject all points yielding a lower bound on the number of inliers. Assuming that we repeat this for each correspondence we get a complexity of $\mathcal{O}(n^2 \log n)$.

Algorithm 1 Fast Outlier Rejection

Given a lower bound, L, on the number of inliers, compute an upper bound B_0 assuming that correspondence 0 is an inlier. If $B_0 < L$ remove correspondence 0.

For each $i \neq 0$
  Compute $M_i \supset \bar{Q}_i - \bar{Q}_0$.
  Find an interval of angles such that $Rv_i + t - q_i \in M_i$.
Sort the set of interval boundaries.
For each interval boundary $\theta^{(i)}$
  Let $b^{(i)}$ be the number of intervals containing $\theta^{(i)}$.
If $\max b^{(i)} + 1 < L$, then remove correspondence 0.

3.3. Technical Details

The starting point for the fast outlier rejection algorithm is an interval of possible heights for the camera. Naturally a shorter interval will allow us to remove more outliers. The full interval is given by estimating the ground plane of the 3D model and assuming that the image was captured between 0 and 10 metres over the ground. This relatively large interval is divided into $k = 10$ subintervals and the outlier rejection is performed for each of these intervals. Correspondences which are rejected at all heights can be permanently removed from the problem. If we have k height intervals we get a total complexity of $\mathcal{O}(kn^2 \log n)$.

4. Outlier Minimization

As described in Section 2.2 we can find the optimal number of inliers by extracting all the KKT points to a constructed optimization problem. First we decide on a goal function f on the parameter space. Normally a linear goal function will yield the simplest equations. For $k = 1, \ldots, 4$ we need to solve the following problem:

> Given a subset S of k residuals compute all points satisfying $r_i = \epsilon$ for $r_i \in S$ such that the gradients of f, the residuals in S and the embedding constraints are linearly dependent.

We will need a specialized solver for each k. One of the solution points generated in this way will be optimal with respect to the number of outliers.

The residual constraint for a point U_i can be formulated as

$$U_i'^T C_i U_i' = 0, \quad U_i' = RU + t. \tag{12}$$

For our application, each of these problems can be formulated as the solution to a system of polynomial equations. We will briefly describe how we construct the first two solvers. We have found that the 1- and 2-point solvers are of little importance in practice, so we do not describe them here.

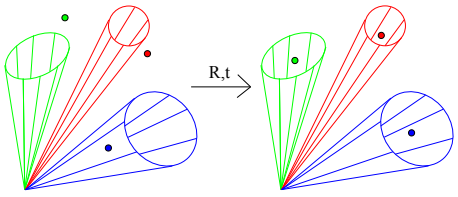

Figure 5. The registration problem for points lying on cones: Find a 3D translation and a planar rotation so that the 3D points lie on or within the cones.

4.1. The 4-Point Solver

The parameter space can be embedded in $\mathbb{R}^5$ by setting

$$\bar{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \tag{13}$$

and adding the embedding constraint $a^2 + b^2 = 1$. The first four equations from (12) are in general full second degree polynomials in the five variables (a, b, t_x, t_y, t_z). Together with the embedding constraint this yields a system of five quadratic equations, which can be solved with the techniques from [4, 7]. This typically yields 28 solutions, but rarely more than 8 real-valued ones. We have implemented a solver where the most time consuming step is doing a QR factorization of a 280×252 matrix. On a desktop computer the running time for this type of solver is in the order of a few milliseconds.

4.2. The 3-Point Solver

Although the technique from [8] is based on introducing a dummy goal function, this function is actually never used in the 4-solver. This is not the case for the 3-solver. To get as simple equations as possible we use a linear goal function, $f = a$, so that $\nabla f = [1\,0\,0\,0\,0]$. This should be linearly dependant with the gradients of the two registration constraints (12), and the gradient of the embedding constraint. This gives a third degree polynomial in the five variables. Combining this equation with the three registration constraints (12) and the embedding constraint we end up with a set of equations that in general give 40 solutions. Again the most time consuming step in the solver is doing a QR factorization, in this case of a 1260×1278 matrix.

4.3. Computational Complexity

Algorithm 2 shows the steps of the outlier minimization algorithm. As the number of sets of ≤ 4 residuals is $\mathcal{O}(n^4)$ we can only do exhaustive sampling of these sets for relatively low number of correspodences. This might seem very restrictive, but first note that as the fast outlier rejection method normally removes all but a few of the outliers, it is

the number of inliers which will be relevant with respect to efficiency. Moreover, in cases with hundreds of inliers, we will get a good pose estimation even without computing the globally optimal one. If speed is prioritized over optimality, another choice is to use ordinary minimal solvers after the rejection step.

For the experiments, we will sometimes exhaustively search all the subsets and sometimes stop when we have detected a good-enough solution (i.e. with enough number of inliers) or reached a maximum number of iterations.

Algorithm 2 Outlier minimization

Given a set of image points u_i and 3D points U_i estimate the pose that minimizes the number of outliers.

Transform the problem to a point-to-cone registration.
For each subset of correspondences of size ≤ 4:
  Use the relevant solver to estimate the KKT points.
  For each KKT point (R, t):
    Count the inliers and update the best solution.

5. Experiments

We have conducted a number of experiments, both on synthetic and real data, to test robustness, speed and accuracy of the proposed methods.

5.1. City-Scale Localization

To evaluate on challenging real-world data, we have performed a localization experiment on the Dubrovnik dataset [19]. It consists of a 3D model with around 2 million points reconstructed from 6000 images. Naturally, each point is also equipped with a SIFT descriptor. Except the 3D model, the dataset also provides 800 test images with computed estimates of camera positions and orientations. As these estimates are also based on vision algorithms, they are not quite ground-truth. In fact they contain some outliers.

When building such a 3D model it is possible to also get a rough estimate of the ground plane and, based on the estimated matchings, get an interval of possible heights for Algorithm 1. As the ground plane is not available for the Dubrovnik dataset we synthesize this information by picking a ± 5 metres interval around the provided estimated height. Note that the length of this interval will mainly affect the running time and not the accuracy of the final result.

A similar problem is that the dataset contains no orientation measurements. Again we synthesize this information using the provided estimated camera orientations and adding a random rotation distributed uniformly on $[0, 1^\circ]$; see Section 2.3 for a motivation.

For each image, correspondences to the 3D model were established using standard SIFT matching (with matching ratio 0.9). Then the image was localized by running Algorithm 1 followed by Algorithm 2 with a maximum of 1000 iterations. As discussed in Section 4.3, we stop early if a reasonable amount of inliers is found.

Table 1 shows a comparison to other methods. In accordance with [19], an image is considered correctly localized if at least 12 correct inliers are found. For the two images where this was not the case, we found 8 and 9 inliers respectively and the errors were small. So essentially these two cases were not failures. Moreover, as our algorithm is optimal for a given bound on the errors, in our case 6 pixels, we can say that for these two images there does not exist a solution with 12 inliers. This is not a contradiction to [18] as we have a more restricted camera model. Also, since there are no correspondences provided in the dataset between the SIFT-points in the query images and the model points, it is somewhat difficult to compare performance between the different methods.

The median error of our method is significantly lower than for the other methods. This shows the advantage of using measurements from an orientation sensor—even if that sensor has an error of up to 1°.

Method	# reg. images	Median error (m)	# error < 18.3 m	# error $> 400m$
Our	798	0.56	771	3
[24]	795.5	1.4	704	9
[23]	783.9	1.4	685	16
[23]	782.0	1.3	675	13
[19]	753	9.3	655	-
[5]	789	-	-	-
[18]	800	-	-	-

Table 1. Results on the Dubrovnik dataset; see [24].

Using a single threaded C-implementation, the median running time for Algorithm 1 was 5.06 seconds, containing 4766 point correspondences, so most problems for this dataset are large. For a more reasonably sized problem of 1000 correspondences, the running time was approximately 0.3 seconds. For the largest problem, with 17199 points, the execution time was 55.4 seconds. For Algorithm 2, we only had a Matlab-implementation. With this implementation, each iteration takes approximately 0.3 seconds. In almost all cases for this experiment, the first step finds more than enough inliers for a very good solution, making it unnecessary to run the second step.

5.2. Shopping Street Experiment

From 412 images, a 3D model was built using the method in [21]. A set of 101 Iphone images from the same street were localized in the 3D model using Algorithm 1

and 2. The gravitational vector was captured by the internal sensors in the phone. One purpose of this experiment is to show that the measured gravitational sensor is indeed accurate enough to use for localization. The experiment also shows that very high rates of outliers can occur in practice. Due to a significant difference in lighting conditions, the feature matching was unusually difficult; see Figure 7. To get any correct matches the SIFT matching ratio was increased to 0.95. Naturally this produced a lot of erroneous correspondences; see Figure 6.

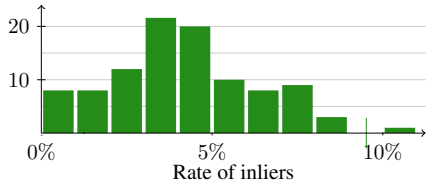

Figure 6. Rate of inliers for the 101 query images in the shopping street experiment.

The results are evaluated by counting the number of inliers, and by visually verifying the inlier correspondences as well as the position on the street. In all cases the computed camera pose was visually correct and in 100 of the images there were at least 12 inliers. In the last case there were only 10 inliers but the pose was still correct.

Figure 7. One of the SLR images used for building the model (left) and one of the Iphone test images (right). Note the illumination difference.

5.3. Semi-Synthetic Experiment

Since we have no way of obtaining ground-truth positions for the shopping street experiment, we also constructed a semi-synthetic setup. The same 3D model as in the shopping street experiment was used and the same correspondences. However, the image points were recomputed to have control over the noise. For each image, a subset of 10 correct image points was selected. Gaussian noise with standard deviation 0.005 was added to the calibrated points and the gravitational vector was corrupted with a uniform noise angle on $[0, 1^\circ]$.

Both for 3-point RANSAC, 2-point (with known vertical direction [15]) RANSAC and Algorithm 2, exhaustive sampling of all the minimal subsets was performed. The localization errors in metres for the three methods are compared in Figure 8. In most cases the methods works well, but the RANSAC methods are more likely to produce large errors.

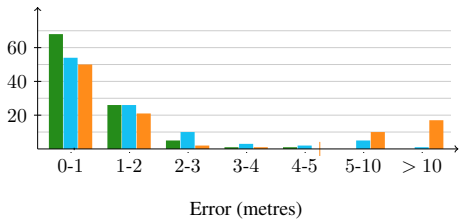

Figure 8. Histogram over the localization error of the proposed method (green) compared to exhaustive 2-point and 3-point RANSAC (blue and orange) for the semi-synthetic experiment.

5.4. Timing Comparison

Another experiment was performed on a subset of the test images from the Dubrovnik dataset. Taking a number of images, all but 10 inliers were removed. Then Algorithm 1 was run with varying number of outliers. The execution times as a function of the number of outliers can be seen in Figure 9. As a comparison, we have run a 3-point pose solver (implemented in C) and a 2-point (plus up-direction) pose solver (implemented in Matlab), in RANSAC loops. The number of RANSAC iterations was chosen such that the probablity of getting at least one outlier-free minimal set was 0.99. Algorithm 1 is much faster than RANSAC for high rates of outliers. This is expected as our outlier removal runs in $\mathcal{O}(n^2 \log n)$ compared to 3-point RANSAC which increases as $\mathcal{O}(n^4)$ for this experiment. Comparing execution times to the 2-point RANSAC is unfair since the implementation used is very inefficient. Simulations of a C-implementation indicate that the solver is approximately as fast as Algorithm 1; being faster for low rates of outliers and slightly slower for higher rates.

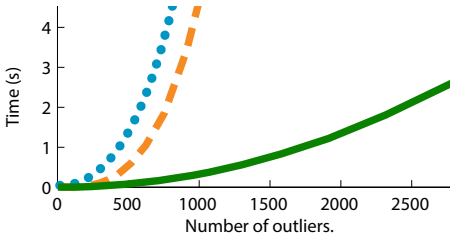

Figure 9. Execution times for a 3-point RANSAC loop (orange dashed line), a 2-point RANSAC loop (blue dotted line) and the proposed outlier removal step (green line). The x-axis shows the number of outliers. The number of inliers was fixed to 10.

6. Conclusions

We have in this paper presented a pose estimation framework that can handle large amount of outliers in the data. It assumes knowledge about the orientation of the camera relative to the ground plane. This information is readily available for many practical applications using e.g. cameras mounted on vehicles or hand held devices such as smart phones with gravitational sensors. The experiments show that using this information we significantly improve both localization accuracy and robustness to outliers.

References

[1] Lis331dlh mems motion sensor. http://www.st.com, 2013. 3

[2] E. Ask, O. Enqvist, and F. Kahl. Optimal geometric fitting under the truncated l_2-norm. In *Conf. Computer Vision and Pattern Recognition*, 2013. 2, 3

[3] T. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 2003. 2

[4] M. Byröd, K. Josephson, and K. Åström. Fast and stable polynomial equation solving and its application to computer vision. *Int. Journal of Computer Vision*, 2009. 5

[5] S. Choudhary and P. Narayanan. Visibility probability structure from sfm datasets and applications. In *European Conf. on Computer Vision*, 2012. 1, 6

[6] O. Chum and J. Matas. Optimal randomized ransac. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2008. 2

[7] D. A. Cox, J. Little, and D. O'shea. *Using algebraic geometry*, volume 185. Springer Verlag, 2005. 5

[8] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *European Conf. on Computer Vision*, 2012. 2, 3, 5

[9] 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. *European Conf. on Computer Vision*, 2010. 2

[10] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim. Pose estimation from corresponding point data. *Systems, Man and Cybernetics*, 1989. 1

[11] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *Conf. Computer Vision and Pattern Recognition*, 2008. 1

[12] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *Conf. Computer Vision and Pattern Recognition*, 2009. 1

[13] F. Kahl and R. Hartley. Multiple view geometry under the L_∞-norm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2008. 2

[14] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007. 2

[15] Z. Kukelova, M. Bujnak, and T. Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *Asian Conf. Computer Vision*, 2010. 2, 7

[16] H. Li. A practical algorithm for L_∞ triangulation with outliers. In *Conf. Computer Vision and Pattern Recognition*, 2007. 2

[17] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Int. Conf. Computer Vision*, 2009. 2

[18] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *European Conf. on Computer Vision*, 2012. 1, 6

[19] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conf. on Computer Vision*, 2010. 1, 6

[20] O. Naroditsky, Z. Zhu, A. Das, S. Samarasekera, T. Oskiper, and R. Kumar. Videotrek: A vision system for a tag-along robot. In *Conf. Computer Vision and Pattern Recognition*, 2009. 2

[21] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. In *Scandinavian Conf. on Image Analysis*, 2011. 6

[22] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *Conf. Computer Vision and Pattern Recognition*, 2010. 2

[23] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Int. Conf. Computer Vision*, 2011. 1, 6

[24] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *European Conf. on Computer Vision*, 2012. 1, 6

[25] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Conf. Computer Vision and Pattern Recognition*, 2007. 1

[26] K. Sim and R. Hartley. Removing outliers using the L_∞-norm. In *Conf. Computer Vision and Pattern Recognition*, 2006. 2

[27] B. Steder, G. Grisetti, S. Grzonka, C. Stachniss, A. Rottmann, and W. Burgard. Learning maps in 3d using attitude and noisy vision sensors. In *Intelligent Robots and Systems*, 2007. 2

[28] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter. An adversarial optimization approach to efficient outlier removal. In *Int. Conf. Computer Vision*, 2011. 2