A Minimal Case Solution to the Calibrated Relative Pose Problem for the Case of Two Known Orientation Angles*

Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys

Computer Vision and Geometry Lab
Department of Computer Science
ETH Zürich, Switzerland
{fraundorfer,marc.pollefeys}@inf.ethz.ch,
tpetri@student.ethz.ch

Abstract. It this paper we present a novel minimal case solution to the calibrated relative pose problem using 3 point correspondences for the case of two known orientation angles. This case is relevant when a camera is coupled with an inertial measurement unit (IMU) and it recently gained importance with the omnipresence of Smartphones (iPhone, Nokia N900) that are equipped with accelerometers to measure the gravity normal. Similar to the 5-point (6-point), 7-point, and 8-point algorithm for computing the essential matrix in the unconstrained case, we derive a 3-point, 4-point and, 5-point algorithm for the special case of two known orientation angles. We investigate degenerate conditions and show that the new 3-point algorithm can cope with planes and even collinear points. We will show a detailed analysis and comparison on synthetic data and present results on cell phone images. As an additional application we demonstrate the algorithm on relative pose estimation for a micro aerial vehicle's (MAV) camera-IMU system.

1 Introduction

In this paper we investigate the case of computing calibrated relative pose for the case of two known orientation angles. This case is largely motivated by the availability of Smartphones (e.g. iPhone, Nokia N900) that are equipped with a camera and an inertial measurement unit (IMU). In the case of Smartphones the IMU mainly consists of accelerometers that allow to measure the earths gravity vector. From this measurement two orientation angles of the device and the embedded camera can be measured but usually not all three orientation angles. A similar situation arises when one is detecting vanishing points in the image. From a detected vanishing point it is also possible to compute two orientation angles [3], and we have the same case as with the accelerometer.

* This work was supported in parts by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant n.231855 (sFly) and by the Swiss National Science Foundation (SNF) under grant n.200021-125017.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 269–282, 2010.

For the relative pose problem this means that only one rotation angle and the three translation parameters are left to be computed from visual measurements. For this special case we will derive a simplified essential matrix and we will show that this leads to an algorithm that can compute the relative pose from three point correspondences only (similar to [5]), instead of the standard 5 point algorithm [10]. In analogy to the 5-point (6-point), 7-point, and 8-point algorithm for computing the essential matrix in the unconstrained case, we derive a 3-point, 4-point, and 5-point algorithm for the special case of two known orientation angles.

Reducing the number of point correspondences is of utmost importance when a RANSAC [1] scheme is used to cope with outliers in the data. The number of random samples to find one outlier free sample depends exponentially on the number of parameters to instantiate one hypothesis. The necessary number of samples to get an outlier free sample with a chance of 99% and an outlier ratio of 50% is 146 for the standard 5-point algorithm. The 3-point algorithm would only need 35 samples which is a speedup of a factor of 4. In R-RANSAC [8], where the termination criterion is, when the probability of missing a set of inliers larger than the largest support found so far, falls under a predefined threshold, a smaller sample size also improves the efficiency. In this sense the proposed 3-point algorithm will be much more efficient in computing relative pose than previous methods, which might be very important for using the method on Smartphones with limited computational power.

We also analyze the degeneracies of the new algorithm and we will show that it will work for planar scenes and even for the case of three collinear points.

In addition to a detailed description of the proposed algorithm, we will give a detailed performance analysis using synthetic data. We will test the algorithm under different levels of noise and more important under noise on the IMU measurements. For results on real data we show relative pose estimation on images from a Nokia N900 Smartphone. The IMU values of the N900 have a precision of 1 degree and we will show that this is good enough to be used for our algorithm. As an additional application we demonstrate the algorithm on relative pose estimation for the camera-IMU system of a micro aerial vehicle (MAV). Visual localization for MAV's is a very active field of research and the power and weight limitations of a MAV do not allow for the use of powerful computers. Therefore it is even more important to use very efficient algorithms, e.g. our proposed 3-point algorithm.

2 Related Work

The special case of knowing the full camera orientation is the case of pure translational motion. In this case the essential matrix can be computed linearly [3]. The case of knowing the translation and solving for the rotation only, using three points, was briefly described in [11]. The case of knowing the rotation partially, e.g. from vanishing points has already been investigated, e.g. for removing tilt in photographs [2] or rectifying images for visual localization [16]. However, the knowledge of two rotations can directly be used to derive a simplified essential matrix that can be estimated from 3 point correspondences and the two rotation angles. This 3-point method has recently been investigated in [5] and this work is closely related to ours. They set up a polynomial equation system in the entries of the simplified essential matrix and use the Macaulay matrix method to solve it which gives 12 solutions for the essential matrix.

In our work we use a similar parameterization of the essential matrix, however we follow a different way of setting up and solving the polynomial system for the essential matrix. The method we propose leads to a 4th degree polynomial which results in up to 4 real solutions for the essential matrix, instead of 12 as in [5]. In terms of efficiency this is an important fact. The different essential matrices have to be verified with additional points (usually done within a RANSAC loop). In this sense our formulation with 4 solutions is much more efficient than the one with 12 solutions.

Our formulation is related to the 5-point algorithm [10] and to the 6-point algorithm [14] in the way the 4th degree polynomial is set up. In addition we also propose a linear 5-point algorithm and a 4-point algorithm which are analogies to the 8-point [6] and 7-point [4] algorithm for computing the essential matrix.

Other approaches that make use of IMU measurements perform Kalman filter fusion of the IMU measurements and the vision based pose estimates [9,12]. In our approach we propose a very tight coupling between IMU and vision measurements in a way that the IMU measurements simplify the vision based camera pose estimation. Another example of tight integration of IMU and visual measurements has also been proposed in [17]. IMU measurements have also been used in [15] together with GPS for large scale structure-from-motion. In this approach GPS and IMU values have been used as initial values for bundle adjustment.

3 Estimating the Essential Matrix for the Case of Two Known Orientation Angles

In this section we derive the parameterization of the essential matrix for the case of two known orientation angles. We identify additional linear constraints in the parameters of E which make it possible to compute a minimal solution for the essential matrix from 3-point correspondences. We will derive this algorithm in detail and give also a 4-point algorithm and a linear 5-point algorithm.

We will start with the definition of a simplified essential matrix where two rotations (pitch and roll) are zero. The essential matrix E can be written as a product of the translation and the three rotation matrices as follows:

$$E = [t]_\times (R_Y R_P R_R) \tag{1}$$

R_Y is the rotation matrix for the yaw axis, R_P is the rotation matrix for the pitch axis and R_R is the rotation matrix for the roll axis. $[t]_\times$ is the skew symmetric matrix form of the translation vector $t = (t_x, t_y, t_z)$.

$$R_Y = \begin{bmatrix} \cos(Y) & \sin(Y) & 0 \\ -\sin(Y) & \cos(Y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$R_R = \begin{bmatrix} \cos(R) & 0 & \sin(R) \\ 0 & 1 & 0 \\ -\sin(R) & 0 & \cos(R) \end{bmatrix} \tag{3}$$

$$R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(P) & \sin(P) \\ 0 & -\sin(P) & \cos(P) \end{bmatrix} \tag{4}$$

With roll and pitch values zero the matrices R_P and R_R are reduced to identity matrices and the essential matrix E gets $E = [t]_\times R_Y$. This expanded gives the simplified essential matrix as written in (5).

$$E = \begin{bmatrix} t_z \sin(Y) & -t_z \cos(Y) & t_y \\ t_z \cos(Y) & t_z \sin(Y) & -t_x \\ -t_y \cos(Y) - t_x \sin(Y) & t_x \cos(Y) - t_y \sin(Y) & 0 \end{bmatrix} \tag{5}$$

By looking at the essential matrix in the form of (5) we can identify 3 linear relations.

$$E = \begin{bmatrix} E_{1,1} & E_{1,2} & E_{1,3} \\ E_{2,1} & E_{2,2} & E_{2,3} \\ E_{3,1} & E_{3,2} & E_{3,3} \end{bmatrix} \tag{6}$$

$$E_{3,3} = 0 \tag{7}$$

$$E_{1,2} = -E_{2,1} \tag{8}$$

$$E_{1,1} = E_{2,2} \tag{9}$$

Using these relations the essential matrix E can be expressed with 6 of its matrix entries reducing the degrees-of-freedom from 8 (up to scale) to 5.

$$E = \begin{bmatrix} E_{2,2} & -E_{2,1} & E_{1,3} \\ E_{2,1} & E_{2,2} & E_{2,3} \\ E_{3,1} & E_{3,2} & 0 \end{bmatrix} \tag{10}$$

In the following we will use the representation of (10) to solve for the essential matrix from point correspondences. The epipolar constraint $\mathbf{x}'^T E \mathbf{x} = 0$ can be written as shown in (11) for the 6 entries of E as written in (12).

$$[x' \;\; xy' - x'y \;\; yy' + xx' \;\; y' \;\; x \;\; y]E = 0 \tag{11}$$

$$E = [E_{1,3} \;\; E_{2,1} \;\; E_{2,2} \;\; E_{2,3} \;\; E_{3,1} \;\; E_{3,2}] \tag{12}$$

Stacking the constraint rows (11) leads to an equation system of the form

$$AE = 0, \tag{13}$$

where each point correspondence contributes one row. The essential matrix has also to fulfill two internal constraints, the $det(E) = 0$ constraint (14) and the trace constraint (15).

$$det(E) = 0 \tag{14}$$

$$EE^T E - \frac{1}{2} trace(EE^T)E = 0 \tag{15}$$

The condition of pitch and roll being zero can be met by knowing the two rotation angles (e.g. from an IMU or from vanishing points) and transforming the image coordinates from a general pose into one with zero pitch and zero roll angles. This can be done by multiplying the image coordinates of the first frame with a homography transform $H = R_P R_R$ which is seen by writing the three rotations explicitly in the epipolar constraint (16). In (17) it can easily be seen that to remove relative pitch and roll rotations, the necessary transformation is $R_P R_R$.

$$\mathbf{x}'([\boldsymbol{t}]_\times R_Y R_P R_R)\mathbf{x} = 0 \tag{16}$$

$$\mathbf{x}'([\boldsymbol{t}]_\times R_Y)(R_P R_R \mathbf{x}) = 0 \tag{17}$$

3.1 The Linear 5-Point Algorithm

The linear 5-point algorithm is a direct consequence of the epipolar constraints written as (13). The essential matrix E (12) has only 6 parameters and is defined up to scale. Every point correspondence gives a constraint in the form of (11). With 5 of these constraints (13) can be linearly solved for the entries of E. The solution has to be corrected so that E fulfills the $det(E) = 0$ constraint and the trace constraint. This is done be replacing E with E' such that the first two singular values are corrected to be identical. This is in analogy to the 8-point algorithm for computing the essential matrix and fundamental matrix [6]. Similar to the 8-point algorithm the linear 5-point algorithm can be used to find a least squared solution to an over-constrained system (13) if more than 5 point correspondences are used.

Differently to the 8-point algorithm, the case of points in a plane is not a degenerate case for the linear 5-point algorithm. It is shown in [13] that A has rank 6 if all 3D points are in a plane. As the linear 5-point algorithm only needs 5 linearly independent equations this is not a degenerate case.

3.2 The 4-Point Algorithm

The 4-point algorithm is an analogy to the 7-point algorithm for the unconstrained case of computing the essential matrix as described in [4]. It uses the $\det(E) = 0$ constraint for the estimation.

With 4 point correspondences A of the equation system (13) has rank 4. In this case the solution to (13) is a 2-dimensional null space of the form

$$E = aE_1 + E_2. \tag{18}$$

The two-dimensional null space E_1 and E_2 can be computed using SVD. The scalar parameter a can be computed using the $\det(E) = 0$ constraint. Expanding the expression $\det(aE_1 + E_2) = 0$ leads to a cubic polynomial in a. There will be one or three solutions for a (complex solutions are discarded) which leads to one or three solutions for E by back-substitution into (18). The algorithm can be used for more than 4 point correspondences. In this case the null space E_1, E_2 is computed in the least squares sense from all the point correspondences.

3.3 The 3-Point Minimal Case Algorithm

The 3-point minimal case algorithm is an analogy to the 5-point minimal case solver [10] and as well to the 6-point minimal case solver [14] for the unconstrained case. It is minimal as it solves for the remaining 3 DOF (up to scale) using only 3 point correspondences. Similar to the 5-point and 6-point methods it uses the trace constraint and the $\det(E) = 0$ constraint to set up a polynomial equation system and to find the solution to E by solving it.

With 3 point correspondences the matrix A of (13) has rank 3. In this case the solution to (13) is a 3-dimensional null space of the form

$$E = aE_1 + bE_2 + E_3 \tag{19}$$

This is the same case as for the 6-point algorithm, where the solution also has the form of a 3-dimensional subspace.

To derive the solution we start by substituting (19) into the $\det(E) = 0$ constraint (14) and into the trace constraint (15), which gives 7 polynomial equations in a and b of degree 3. Our reduced essential matrix parameterization (12) has only 6 parameters, thus the trace constraint gives only 6 equations instead of 9. And the seventh equation comes from the $\det(E) = 0$ constraint. However the rank of the linear system of the 7 equation is only 6. There exists a linear dependency between the entries $E_{1,3}, E_{2,3}, E_{3,1}, E_{3,2}$ which can be verified by symbolic Gaussian elimination on the equations.

Considering this as a homogeneous linear system in the monomials of a and b this will give expressions in the monomials $a^3, a^2b, a^2, ab^2, ab, a, b^3, b^2, b, 1$. Next we set up a polynomial equation system using the 6 linearly independent equations. By performing Gaussian elimination and subsequent row operations a polynomial of 4th degree in b can be found. From Gaussian elimination we get (20).

$$\begin{array}{cccccccccccc}
a^3 & a^2b & a^2 & ab^2 & ab & a & b^3 & b^2 & b & 1 & \\
\hline
1 & & & & & & . & . & . & . & \\
 & 1 & & & & & . & . & . & . & \\
 & & 1 & & & & . & . & . & . & \\
 & & & 1 & & & . & . & . & . & \\
 & & & & 1 & & . & . & . & . & < h > \\
 & & & & & 1 & . & . & . & . & < i >
\end{array} \tag{20}$$

With row operations on the polynomials $<h>$ and $<i>$ we can arrive at the desired 4th degree univariate polynomial.

$$<h> = poly(ab, b^3, b^2, b, 1) \tag{21}$$

$$<i> = poly(a, b^3, b^2, b, 1) \tag{22}$$

We eliminate the monomial ab by multiplying $< i >$ with b and subtracting it from $< h >$.

$$< k > = < h > -b < i > = poly(b^4, b^3, b^2, b, 1) \tag{23}$$

The resulting polynomial $< k >$ is a 4th degree polynomial in b. The solutions for b by solving $< k >$ (23) can be substituted into $< i >$ (22) which is linear in a and which gives one solution for a for each b. For each pair of (a, b) we compute an essential matrix using (19).

It can be shown (e.g. with symbolic software like the Gröbner basis package of Maple) that the 3-point problem can have up to 4 solutions. In this sense our solution with the 4th degree polynomial is optimal. The algorithm can also be used for more than 3 point correspondences. In this case the null space E_1, E_2, E_3 is computed in the least squares sense from all the point correspondences.

4 Degeneracies

In [13] it is shown that 6 points in a plane give 6 linearly independent equations of the form of (11), however any further point in the plane will not give another linearly independent equation. This means that algorithms that need more than 6 independent linear equations, e.g. the 7-point and 8-point algorithm will fail for the case of a plane. The 5-point and the 6-point algorithm and the 3-point, 4-point and linear 5-point as well are able to deal with the case of all the points in a plane.

The case of collinear points is another interesting case. It is a degenerate case for the 5-point algorithm but not for the 3-point, which can compute the essential matrix from 3 collinear points. Three points on the same line give three independent equations for the epipolar constraint (11), however any further points on the line will not give another linearly independent equations [13]. For the 3-point algorithm only three linearly independent equations in (11) are needed. The equation system (20) has rank 6 in this case which allows our method to solve the system. However the case where the 3D line is parallel to the baseline of the cameras is a degenerate case. There we observed that the rank of (20) drops to 3.

5 Experiments

5.1 Synthetic Data

In this experiment we evaluate the robustness of the new methods under image noise and noise on the IMU measurements and compare it to the standard 5-point method [11]. Robustness to noise from the IMU is especially important since errors in the two given angles can influence the outcome of the new algorithms.

The test scene consists of random 3D points that have a depth of 50% of the distance of the first camera to the scene. The baseline between the views is 10% of the distance to the scene and the direction is either parallel to the scene (sideways) or along the z-axis of the first camera (forward). Additionally, the second camera was rotated around every axis. This is similar to Nistér's test scene described in [11].

We want to test the algorithms used for two cases, the minimal case when they are used for RANSAC and the least squares case when using many points for a polishing step. For the first series of experiments the minimal number of point correspondences necessary to get a solution is used. If the method resulted in multiple solutions the one with the smallest deviation of the true solution was chosen. For the minimal case experiments 500 trials were done per data point in the plots and the average translation and rotation error of the first quantile are plotted. This measure is a good performance criterion if the method is used for RANSAC where it is more important to find an acceptable solution with many inliers than to get consistent results over all trials. The less accurate solutions will result in less inliers and be automatically filtered out. The least squares plots show the mean value of 200 trials with 100 point correspondences per trial.

To analyze the robustness the computed essential matrices are decomposed in a relative translation direction and rotation. The translational error is the

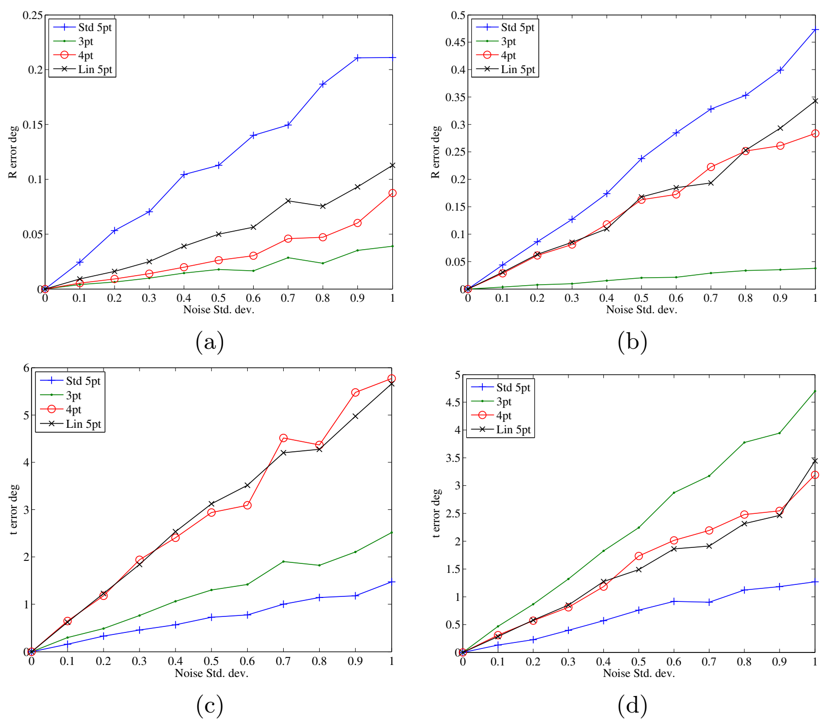


Fig. 1. Rotational and translational error for the minimal case algorithms in degrees over image noise standard deviation in pixel. (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

angle between the true translation direction and the estimated direction. The rotational error is the smallest angle of rotation which is needed to bring the estimated rotation to the true value.

Figure 1 shows the results of the minimal cases for gradually increased image noise levels with perfect IMU data. As can be seen, the new methods produce slightly less robust translation estimates. One reason for that is that with only three resp. four image points the 3-point and 4-point methods are generally more sensitive to image noise. Notice that the 3-point method performs better for sideways motion than for forward motion. Regarding the rotational errors the new methods are more robust than the standard 5-point method, but it has to be noticed that the errors of all algorithms are very small. In addition to that the standard 5-point method has to estimate rotations around 3 axes whereas the new methods estimate only one.

The least squares results in Fig. 2 show that the new methods are as robust as the standard 5-point method for sideways motion. For forward motion, which is the weak point of the standard 5-point algorithm, the 3-point and linear 5-point methods perform better, but the 4-point method does not produce better results.

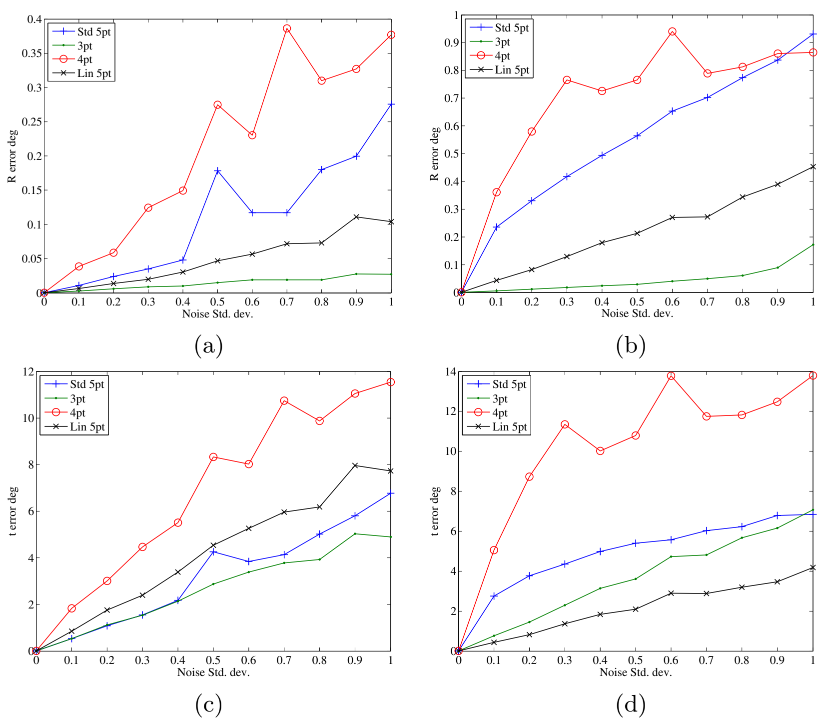


Fig. 2. Rotational and translational error (least squares case) in degrees over image noise standard deviation in pixel. (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

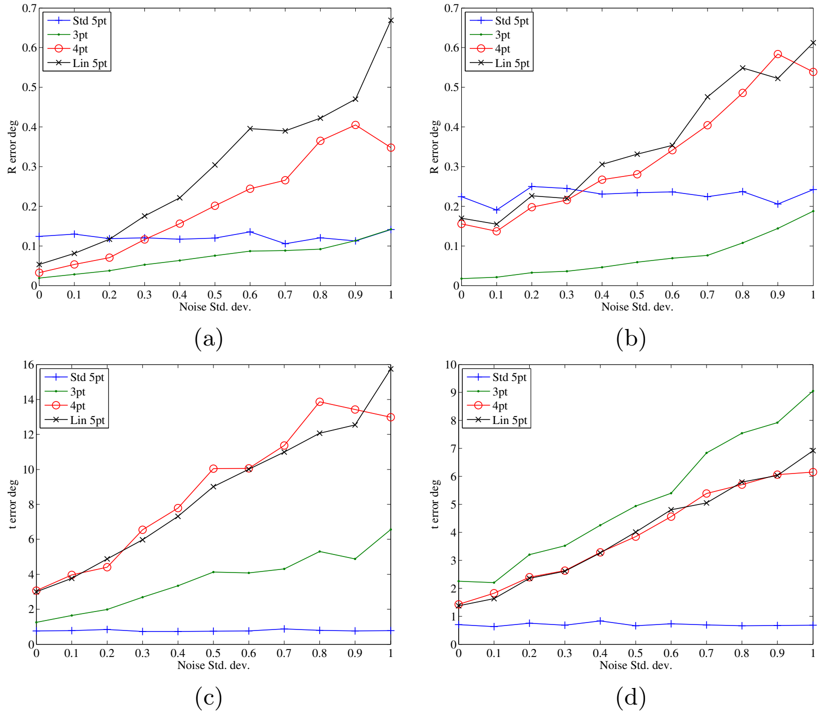


Fig. 3. Rotational and translational error for the minimal case algorithms in degrees over IMU noise standard deviation in degree (at 0.5 pixel image noise). (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

Since in reality no exact data will be available about the attitude of the camera, it is important to examine how the new methods work when noise is added to the IMU data. Good accelerometers today have noise levels of around 0.02 degrees in the computed angles. Figure 3 shows the results for the minimal cases for increasing noise on the IMU data while assuming image noise with 0.5 pixel standard deviation. The standard 5-point method is not influenced by the noise because no IMU data is used. But for the other algorithms the error scales with the noise of the IMU data. This shows that it is important to have good measurements from the IMU because too big errors in the angles make it harder to compute the correct solution. The 3-point method gives acceptable results for sideways motion even for very noisy IMU data. The least squares results in Fig. 4 show a similar picture. For sideways motion and moderate noise the 3-point and the linear 5-point methods are as robust as the standard 5-point method, where for forward motion the linear 5-point is again the most robust of the presented methods. With more image points used for the estimation the new methods benefit from more robustness to image nose, but it is apparent that more image points do not help that much against noise and imprecisions from the IMU.

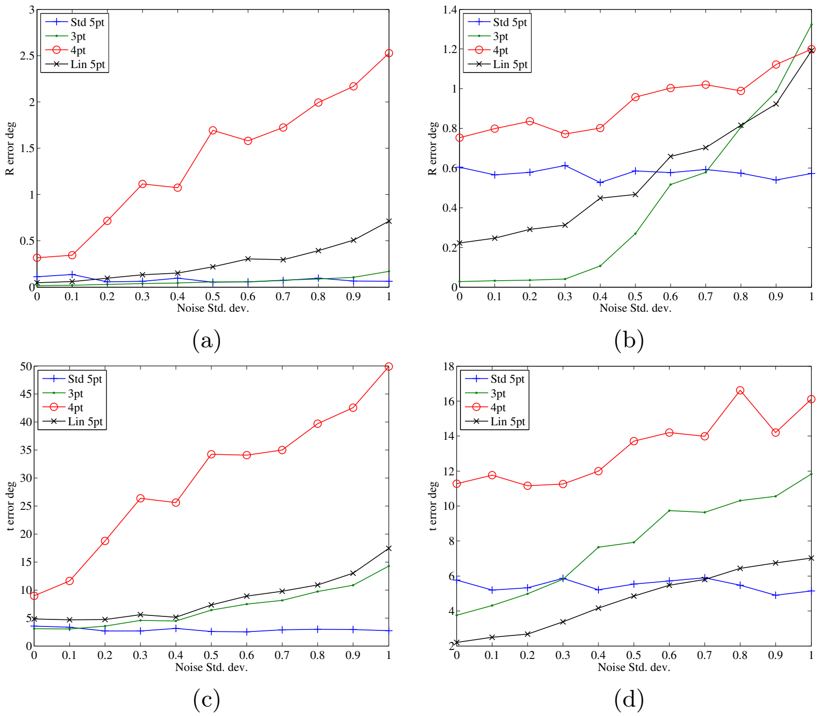
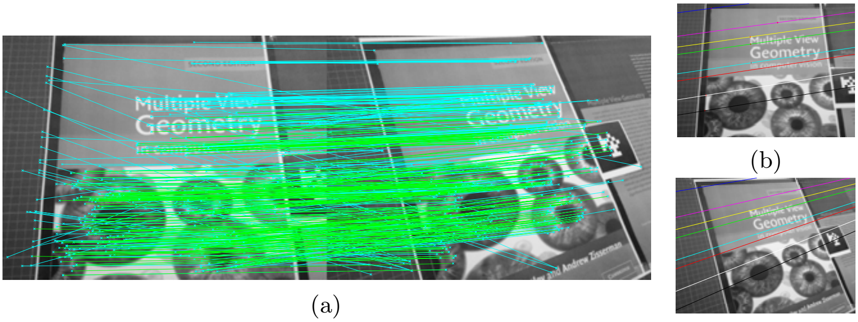


Fig. 4. Rotational and translational error (least squares case) in degrees over IMU noise standard deviation in degree (at 0.5 pixel image noise). (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

The results of the synthetic experiments show that the new methods and especially the 3-point method can be used for example for faster RANSAC to get a set of consistent point correspondences with similar robustness as the standard 5-point method.

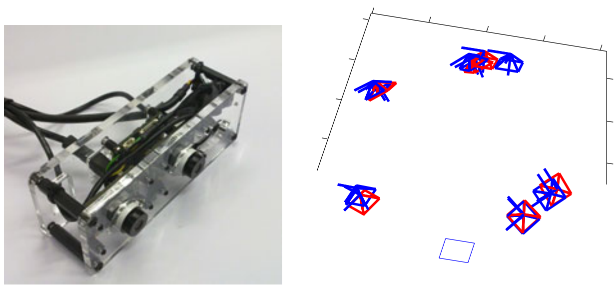
5.2 Real Data from N900 Smartphone

To demonstrate that the new 3-point method also works on currently available consumer Smartphones. We tested it with the Nokia N900 that has an accelerometer similar to other modern Smartphones like Apple's iPhone. The relative rotations to the plane perpendicular to the gravity vector were read out from the accelerometers and used to warp the extracted SIFT [7] features to a virtual ground plane. These transformed feature coordinates only differ in one rotation angle and translation and where used as input for the 3-point method. Fig. 5 shows an example of estimated epipolar geometry for a planar scene. The blue lines show the initial SIFT matches and the green lines the inliers after RANSAC.



(a) (b) (c)

Fig. 5. Epipolar geometry from N900 images and IMU readings. (a) Cyan lines indicate initial SIFT feature matches, green lines are inliers after RANSAC. Residual Sampson distance of inliers is 0.4 pixel. The quality of inliers depends largely on the accuracy of IMU readings. (b,c) Epipolar geometry visualized by epipolar lines.



(a) (b)

Fig. 6. Relative pose estimation for MAV camera-IMU system. (a) The camera-IMU system for a micro quadrotor. (b) Poses computed from the 3-point algorithm (red) compared to marker based pose estimation (blue).

5.3 Relative Pose Estimation for MAV Camera-IMU System

In this experiment we test the 3-point algorithm for relative pose estimation for a MAV camera-IMU system. The camera-IMU system consists of a stereo setup and an IMU with a nominal accuracy of 0.06 degrees. The camera-IMU system is designed to fit on a micro quadrotor. In the experiment we compute the pose of the camera-IMU relative to a reference pose for multiple different poses using SIFT [7] feature matches. To verify the results we also compute the poses using a marker based pose estimation method[1] without using IMU information. The computed poses are plotted in Fig. 6. The poses from the 3-point method are in red color, the poses from the marker based method are in blue. The blue square is the location of the marker on the floor. This experiment successfully demonstrates the practicability of our proposed 3-point method.

[1] AR Toolkit Plus: http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php

6 Conclusion

In this paper we presented three new algorithms (3-point, 4-point and linear 5-point) to compute the essential matrix for the case when two orientation angles are known. We showed that this is a practically relevant case as modern Smartphones (iPhone, Nokia N900) are equipped with an IMU that can measure two of the three orientation angles of the device. This is done by accelerometers which are able to measure the earths gravity normal and thus can measure two of its three orientation angles. This case is also practically relevant for the case of pure visual relative pose estimation for the case when a vanishing point can be detected in the images. Similar to an IMU a single vanishing point also gives two orientation angles. What's more, it is conceivable that in future, cameras will always be coupled with IMU's. The small scale of MEMS IMU's make a tight integration possible. This makes the proposed method even more practically relevant. In the experimental section we analyzed the accuracy of the proposed methods very carefully using synthetic data. A special focus was put on the case of noisy IMU measurements. We showed that the algorithm is sensitive to inaccurate IMU measurements, especially the translation part. However, we showed with real experiments using a Nokia N900 Smartphone, that the accuracy of $1°$ is sufficient for robust relative pose estimation. More experiments conducted with an IMU with higher precision ($0.06°$) showed how the proposed methods can reliably be used for relative pose estimation of an micro aerial vehicle. So far we did not explore the possibility of using the two orientation angles to unwarp the images before feature detection and matching. By unwarping, the perspective distortions between the two images could be removed. This would allow the use of simpler feature detectors, which need to be only scale and rotation invariant.

References

1. Fischler, M.A., Bolles, R.C.: RANSAC random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of ACM 26, 381–395 (1981)
2. Gallagher, A.: Using vanishing points to correct camera rotation in images. In: Computer and Robot Vision, pp. 460–467 (2005)
3. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge (2000)
4. Hartley, R.: Projective reconstruction and invariants from multiple images. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(10), 1036–1041 (1994)
5. Kalantari, M., Hashemi, A., Jung, F., Guédon, J.P.: A new solution to the relative orientation problem using only 3 points and the vertical direction. arXiv 0905.3964 (2009)
6. Longuet-Higgins, H.: A computer algorithm for reconstructing a scene from two projections. Nature 293, 133–135 (1981)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)

8. Matas, J., Chum, O.: Randomized ransac with sequential probability ratio test. In: Proc. IEEE International Conference on Computer Vision, vol. II, pp. 1727–1732 (2005)
9. Naroditsky, O., Zhu, Z., Das, A., Samarasekera, S., Oskiper, T., Kumar, R.: Videotrek: A vision system for a tag-along robot. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1101–1108 (2009)
10. Nistér, D.: An efficient solution to the five-point relative pose problem. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. II195–II202 (2003)
11. Nistér, D., Schaffalitzky, F.: Four points in two or three calibrated views: theory and practice. International Journal of Computer Vision, 211–231 (2006)
12. Oskiper, T., Zhu, Z., Samarasekera, S., Kumar, R.: Visual odometry system using multiple stereo cameras and inertial measurement unit. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
13. Philip, J.: Critical point configurations of the 5-,6-,7-, and 8-point algorithms for relative orientation. In: Technical Report TRITA-MAT-1998-MA-13, KTH (1998)
14. Pizarro, O., Eustice, R., Singh, H.: Relative pose estimation for instrumented, calibrated imaging platforms. In: Proceedings of the Seventh International Conference on Digital Image Computing: Techniques and Applications, pp. 601–612 (2003)
15. Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. International Journal of Computer Vision 78(2-3), 143–167 (2008)
16. Robertsone, D., Cipolla, R.: An image-based system for urban navigation. In: British Machine Vision Conference, pp. 1–10 (2004)
17. Steder, B., Grisetti, G., Grzonka, S., Stachniss, C., Rottmann, A., Burgard, W.: Learning maps in 3d using attitude and noisy vision sensors. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 644–649 (2007)