Trust Your IMU: Consequences of Ignoring the IMU Drift

Marcus Valtonen Örnhag¹, Patrik Persson¹, Mårten Wadenbäck², Kalle Åström¹, Anders Heyden¹

¹Centre for Mathematical Sciences
Lund University

²Department of Electrical Engineering
Linköping University

marten.wadenback@liu.se

Abstract

In this paper, we argue that modern pre-integration methods for inertial measurement units (IMUs) are accurate enough to ignore the drift for short time intervals. This allows us to consider a simplified camera model, which in turn admits further intrinsic calibration. We develop the first-ever solver to jointly solve the relative pose problem with unknown and equal focal length and radial distortion profile while utilizing the IMU data. Furthermore, we show significant speed-up compared to state-of-the-art algorithms, with small or negligible loss in accuracy for partially calibrated setups.

The proposed algorithms are tested on both synthetic and real data, where the latter is focused on navigation using unmanned aerial vehicles (UAVs). We evaluate the proposed solvers on different commercially available low-cost UAVs, and demonstrate that the novel assumption on IMU drift is feasible in real-life applications. The extended intrinsic auto-calibration enables us to use distorted input images, making tedious calibration processes obsolete, compared to current state-of-the-art methods. Code available at: https://github.com/marcusvaltonen/DronePoseLib.

1. Introduction

A popular approach in Simultaneous Localization and Mapping (SLAM) is to fuse various sensor data to increase the performance of the system. A common pair of sensors to combine is a camera and an IMU. Systems of this kind are labeled as visual-inertial odometry, and this specific sensor combination is often found on consumer devices, such as smartphones and UAVs. As is well-known, the projective relationship between two cameras manifests itself in the fundamental matrix, independent of the scene geometry. When auxiliary data are known (e.g. IMU data) the number of degrees of freedom decreases and the corresponding fundamental matrix is constrained, which enables one to compute it with fewer point correspondences. This potentially reduces the impact of noise; however, such algebraic constraints can be highly non-trivial to incorporate in a solver. Popular methods to handle the resulting polynomial systems of equations use theory from algebraic geometry, e.g. the action matrix method [3], turning the problem...
into a generalized eigenvalue problem (GEP) \[19, 21\], and resultant based methods \[1\]. Regardless of which method is used, there is still much work in terms of parameterizing the original problem, as different approaches may yield completely different results \[24\]. For the resulting solver to be of any practical use, it must also be numerically stable, which adds further considerations to the design.

Since modern image sensors often have square-shaped pixels and the lens is sufficiently aligned such that the principal point coincides with the optical center, a feasible assumption is to use partially calibrated cameras, where the only unknown intrinsic parameter is the focal length. This specific assumption has proven useful in several different real scenarios including relative pose estimation \[6, 12, 14, 18, 23\] and absolute pose estimation \[26, 42\].

Although it often comes at the cost of introducing distortion, having a wide field of view is desirable in many applications. When working with visual odometry, it is, therefore, a standard procedure to correct for these undesirable distortion artifacts, which often requires a specific calibration setup, typically involving a checkerboard pattern. By incorporating a distortion model, as well as focal length, together with the motion model, one may omit such procedures altogether; however, due to the difficulty of the problem, no fast and robust minimal solver has yet been proposed. The main contributions of this paper are:

- We take advantage of IMU data to estimate the full 3D orientation. Under the assumption that the IMU drift is negligible for short time intervals, the resulting polynomial systems of equations are significantly easier to solve.

- By using this approach, we are able to treat the partially calibrated case with unknown radial distortion profile while incorporating the IMU data, resulting in a fast and reliable solver. This is the only solver to handle this case to date.

- Furthermore, we show a considerable speed-up compared to other state-of-the-art methods, with small or insignificant loss in accuracy, when exploiting the assumption of negligible IMU drift. This benefits low-cost and embedded devices, which constitute the majority of consumer devices where these algorithms are used in practice.

2. Previous Work

2.1. Visual-Inertial Odometry

The calibrated visual-inertial problem of relative pose is well-studied \[9, 28, 33, 38\] and efficient solvers exist. If we assume that the gravity direction is aligned with the \(y\)-axis, the corresponding essential matrix (after alignment) is given by \(E_y \sim \begin{bmatrix} t \end{bmatrix} \times R_y\), or explicitly,

\[
E_y = \begin{bmatrix}
-t_y \sin \phi & -t_z \cos \phi & -t_y \\
t_x \cos \phi + t_z \sin \phi & 0 & t_z \cos \phi - t_x \cos \phi \\
-t_y \cos \phi & t_x \sin \phi & -t_y \sin \phi
\end{bmatrix},
\]

Figure 2. Assume the IMU measurements are accurate, i.e. the accelerometer and gyroscope data can be used to accurately estimate the relative orientation between two consecutive views. Then the only unknown extrinsic parameter is the translation vector between the poses.

where

\[
R_y = \begin{bmatrix}
\cos \phi & 0 & \sin \phi \\
0 & 1 & 0 \\
-\sin \phi & 0 & \cos \phi
\end{bmatrix},
\]

and \(t = (t_x, t_y, t_z)\). This makes it possible to use a parameterization with six elements,

\[
E_y = \begin{bmatrix}
e_1 & e_2 & e_3 \\
e_4 & e_5 & e_6 \\
-e_3 & e_6 & e_1
\end{bmatrix}.
\]

Since we have four degrees of freedom (three translation elements and one angle), the elements \(e_i\) of (3) are not independent. In fact, one can check that they must obey the (modified) Demazure equations, also known as the trace constraint, \(2E_y E_y^{T} E_y - tr(E_y E_y^{T}) E_y = 0\), or explicitly,

\[
e_1 e_2 e_6 + e_1 e_4 e_5 + e_3 e_5^2 - e_3 e_6^2 = 0,
\]

\[
e_1 e_2^2 - e_1 e_6^2 - e_2 e_3 e_6 + e_3 e_4 e_5 = 0.
\]

as well as the rank constraint \(\det(E_y) = 0\),

\[
e_1 e_2 e_4 + e_1 e_5 e_6 + e_2 e_3 e_5 - e_3 e_4 e_6 = 0.
\]

These constraints were used in \[9\] to build a minimal solver for the calibrated case\(^2\). We also note an easy decomposition into rotation and translation components, given

\(^2\)In \[9\] they align the \(z\)-axis with the gravity instead.
by \( e_1^2 + e_3^2 = t_y^2 \). In [12] it was shown that the minimal case can be solved using a single affine correspondence.

The problem becomes more difficult when adding an unknown focal length. Without any IMU data available, but still considering the partially calibrated case with only unknown focal length, the corresponding fundamental matrix has six degrees of freedom. This problem, therefore, requires a minimal case of six point correspondences, with the current state-of-the-art solver by Kukelova et al. [23]. Ding et al. [6] proposed a minimal solver for two partially calibrated cases while incorporating the IMU data. This was done by explicitly parameterizing the rotation about the gravity direction, and turning the problem into a generalized eigenvalue problem (GEP).

When assuming \( R_y = I \), see Fig. 2, the essential matrix is \( E = [t] \times \), which makes the governing equations significantly easier. The minimal calibrated case requires only two point correspondences, and the epipolar constraint for a single pair of point correspondences \( x \leftrightarrow x' \), is given by

\[
x^T E x = 0 \Leftrightarrow (x \times x')^T t = 0. \tag{6}
\]

In [10] it was also shown that the non-minimal case can be solved with global optimality guarantees.

### 2.2. Relative Pose with Unknown Distortion Profile

When constructing minimal solvers, it is often desirable to use as few parameters as possible. This increases robustness in RANSAC-like frameworks, as fewer iterations are needed in order to select a sample free from outliers. The one-parameter division model [7], has therefore been frequently used, as it performs well with only a single parameter incorporating IMU data. This was done by explicitly parameterizing the unknown distortion parameter, was first studied in [14]; however, no viable real-time solver for the case exists.

The one-sided case, \( i.e. \) with one calibrated camera and one camera with unknown focal length and radial distortion parameter, has been studied further. The first solver was introduced in [18], but was not numerically stable and the elimination template size was quite large, \( 200 \times 231 \). It has later been improved in [23], and is now both numerically stable and fast, with an elimination template size of \( 51 \times 70 \). The one-sided case, however, is mostly artificial, as it assumes one of the cameras to be calibrated, which limits the applicability of the method severely.

To the best of our knowledge, the relative pose problem with unknown and equal focal length and radial distortion parameter incorporating IMU data has not been solved. We will solve this case and show that the resulting solver is extremely fast compared to the methods discussed in this section, with an elimination template size of merely \( 10 \times 21 \). This is done using a special assumption, which we shall discuss next.

### 3. Why Ignore the IMU Drift?

When measurements from the accelerometer and gyroscope are combined in an orientation filter [17, 29, 36, 36, 39] the gravity direction is preserved; however, the yaw angle begins to drift. Because of this, most visual-inertial models try to incorporate an unknown angle about the gravity direction [4–6, 9, 12, 28, 33, 38, 41]. Already in the calibrated case the governing equations (4)–(5) are non-trivial and quadratic or cubic in nature. Parameterizing the rotation matrix will also result in at least second order equations.

Instead of filtering, another approach is to simply integrate the gyroscopic data to obtain the relative orientation [8]. This eliminates error sources that potentially are introduced while fusing the gyroscopic data with the accelerometer data, \( e.g. \) Coriolis forces—when the IMU is not in the center of rotation—or acceleration due to motion. Both corrupt the measurement of the gravity direction. Over time integration errors, sensor bias and sensor noise will lead to drift in all axes; however, for short time intervals, this drift is very small. Additionally, if the IMU sensor has been stationary at some point in time—which is a reasonable assumption for UAVs that start from a ground position—the gyroscopic bias can be observed. The bias changes very slowly, even in low-cost consumer-available IMUs, and the observed bias can therefore be used to compensate for it in the remaining part of the sequence. In summary, we obtain very accurate relative rotations by simply integrating the gyroscopic data between camera pairs.

Valtonen Ornhug et al. [40] proposed using orientation filters to estimate the camera rotation and thereby bene-
fit from a relaxed problem. They showed that this allowed them to perform radial distortion correction, while still maintaining speed and accuracy. The method, however, relied on a homography-based approach, requiring planar objects in the scene geometry, thus limiting the applicability of the method.

Temporarily losing the gravity direction is not a major concern for the relative pose problem. In a complete SLAM framework, one would typically perform a visual-inertial initialization step [2, 15, 30, 32] which recovers the gravity direction as well as metric scale and bias. By trusting the IMU data we note the following:

(i) we remove one degree of freedom from the camera parameterization,
(ii) the relative pose problem becomes linear in the unknown translation, according to (6),
(iii) we open up the possibility for further intrinsic calibration and still perform fast and accurate in real-time applications.

In the next section, we will show how this is done in practice.

4. Consequences of Ignoring the IMU Drift

We construct two solvers based on our simplifying assumption. Note that the derivations are quite short, which is mainly due to the linear dependence on the translation vector.

4.1. Unknown and Equal Focal Length (3-point)

For the case of unknown and equal focal length, the fundamental matrix is given by \( F = K^{-1}[t] \times RK^{-1} \), where \( R = R_{img}^{(2)} R_{im}^{(1)^T} \) is the relative orientation. By parameterizing \( K^{-1} = \text{diag}(1, 1, w) \), we have four unknowns—the translation \( t \) and \( w \). The epipolar constraints \( x_i^T F x_i = 0 \), for \( i = 1, 2, 3 \), yield three equations, which are linear in \( t \). Therefore, the resulting system can be written as

\[
M(w)t = 0,
\]

where \( M(w) \) depends only on \( w \). Even with non-degenerate configurations, the matrix \( M \in \mathbb{R}^{3 \times 3} \) cannot have full rank, hence \( \det(M) = 0 \). This yields a single quartic equation in the unknown \( w \), which can be solved efficiently using the quartic root finding formula. Consequently, we have four putative solutions, from which the translation \( t \) can be retrieved by finding the null space of the \( 3 \times 3 \) matrix \( M(w^*) \), where \( w^* \) is a putative solution. For the \( 3 \times 3 \) case one can do this without resulting to costly SVD computations [11, 31].

4.2. Unknown and Equal Focal Length and Radial Distortion Parameter (4-point)

Assuming the one-dimensional division model (7), the (modified) epipolar constraint is given by

\[
f(x_i, \lambda) = K^{-1}[t] \times RK^{-1} f(x_i, \lambda) = 0,
\]

for a pair of corresponding points \( x_i \leftrightarrow x_i' \), where \( K^{-1} = \text{diag}(1, 1, w) \), with \( w \neq 0 \), and \( \lambda \) is the unknown distortion parameter. As in the previous case, we may utilize the fact that (9) is linear in \( t \), hence

\[
M(w, \lambda)t = 0,
\]

where \( M \in \mathbb{R}^{4 \times 3} \). This can be viewed as seeking the non-trivial nullspace of \( M \), which, in turn, implies that all \( 3 \times 3 \) subdeterminants of \( M \) must vanish. Since there are four such subdeterminants, we can reduce the problem to four polynomial equations in two unknowns, \( w \) and \( \lambda \). Furthermore, we need to exclude non-physical solutions corresponding to \( w = 0 \), as it turns out that there are infinitely many solutions if we allow these. This can be accomplished by saturating the corresponding ideal and can be done using the automatic Gröbner basis generator proposed in [25]. It turns out that the remaining system has eleven solutions in general; however, in practice, the most common case is that 4–6 solutions are real-valued. By using the action matrix method [3], we are able to construct an elimination template of size \( 10 \times 21 \), by using the basis heuristic proposed in [27].
5. Time Complexity

To compare timings in a fair and accurate way all solvers are implemented in C++ using the Eigen [13] library, with the same compilation flags and setup. The only exception is that the original solver by Jiang et al. [14] is in native MATLAB, hence significantly slower. To make a fair comparison we use the slightly faster (but numerically unstable) solver proposed in [24] which is available in C++. Note, therefore, that the real execution time for the original method would be even larger than reported. The faster solver is only used for timing, and the original solver is used for all other experiments.

To simulate a realistic environment we use a Raspberry Pi 4 to measure the execution time, as it is a fair approximation of hardware you can expect on an embedded device running these algorithms. The results are shown in Tab. 1.

Table 1. Mean execution time on a Raspberry Pi 4 for 10,000 randomly generated problem instances in C++. We also show the number of solutions for each problem. These will have to be evaluated (or at least a subset) in a RANSAC-like system, hence affect the total execution time.

<table>
<thead>
<tr>
<th>Author</th>
<th>Time (μs)</th>
<th>No. Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our (3-point)</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Ding et al. [6]</td>
<td>4815†</td>
<td>20</td>
</tr>
<tr>
<td>Kukelova et al. [23]</td>
<td>363</td>
<td>15</td>
</tr>
<tr>
<td>Our (4-point)</td>
<td>1290 ‡</td>
<td>11</td>
</tr>
<tr>
<td>Jiang et al. [14]</td>
<td>1,260,700‡</td>
<td>68</td>
</tr>
</tbody>
</table>

†: C++ implementation received from the authors of [6].‡: Based on the smaller template reported in [24] which is numerically unstable. The original solver would be even slower.

Comparing the proposed 3-point solver to the state-of-the-art solver by Ding et al. [6]—essentially solving the same problem, with the exception that we ignore the potential IMU drift—our solver is more than 800× faster. In addition, the proposed 4-point solver including focal length and radial distortion correction is a factor 3.7× faster than the solver by Ding et al. [6], and significantly faster than the solver by [14]; in fact, it is roughly 1000× faster, bridging the gap from what was considered a theoretically interesting case to something that can be applied in practice.

Let us emphasize the practical implications of simultaneously estimating the distortion parameter: the added intrinsic calibration liberates the user from time-consuming calibration procedures. This allows UAV operators (and those of other visual-inertial systems) to change optics out in the field, with no intermediate setup procedures or specific requirements needed.

6. Synthetic Experiments

In this section, we test the numerical stability and noise sensitivity of our proposed methods compared to the current state-of-the-art. The competing methods are the 4-point solver by Ding et al. [6], the 6-point solver by Kukelova et al. [23] and the 7-point solver by Jiang et al. [14]. We found that the solver proposed in [24], which was reported to have a smaller template size, was numerically unstable; hence we use the original method instead.

In order to get a realistic setup, random synthetic scene points with a positive depth in front of the cameras were generated. Specifically, the scene points \((X, Y, Z)\) were uniformly distributed with \(X, Z \in [-3, 3]\), the depth \(Y \in [3, 8]\), and focal length \(f \in [300, 3000]\). This mimics the setup used in [6]. Furthermore, the orientations are random and facing the scene. The image points are then obtained by projecting the scene points through the cameras, and the orientations of the cameras are used as input to the visual-inertial solvers. To increase numerical stability, the image point correspondences were normalized; this was...
done in the same way for all solvers. The error distribution for noise-free data is shown in Fig. 3. Here, all methods perform well, with a slight advantage for our 3-point method. Note that we include radial distortion for the proposed 4-point solver and the 7-point solver in this case as well; however, the error histograms are similar and therefore omitted.

We proceed by analyzing noise sensitivity in various situations. We test all methods on synthetic data without radial distortion (including the 4-point and 7-point solvers), and add a pixel noise relative to focal length (Gaussian noise with zero mean and standard deviation $1080/f$ pixels, where $f$ is the ground truth focal length). Furthermore, we add noise to the IMU measurements—here we add noise on all angles, including the yaw angle. The noise interval is meant to cover the precision of a low-cost IMU, with a maximal error of approximately $0.5^\circ$, as reported in [20]. The results are presented in Fig. 4. Note that both our methods perform better than the competing methods in terms of fundamental matrix recovery for smaller noise levels, and the state-of-the-art method [6] only performs slightly better for larger noise levels. This is primarily since it is capable of correcting for the error about the gravity direction. We have, however, found empirically on real-data that the lower noise levels are dominating the input data under certain conditions. This holds true even for low-cost devices, hence do not pose a practical issue.

7. Real Data

To demonstrate the applicability of our assumption, and the solvers based on it, we have used the challenging dataset [41] consisting of various indoor and outdoor scenes with predominantly planar surfaces. The scenes were captured using a mid-sized UAV (500 g, $170 \times 240 \times 40$ mm) equipped with a monochrome global shutter camera (OV9281) recorded with resolution $640 \times 400$, equipped with an onboard IMU of model MPU-9250.

In addition, we have recorded a new dataset with a different UAV. The scenes are typically demanding, e.g. an indoor sequence containing forward motion in a corridor, which is known to be hard. We show example images from the scenes in Fig. 5. This dataset uses a lightweight (27 g, $92 \times 92 \times 29$ mm) and low-cost nano quadcopter available under the name Crazyflie 2.0, captured in $640 \times 480$ resolution with an RGB camera (RunCam Nano). The ground truth was obtained using a complete SLAM system, where the reprojection error and IMU error were jointly minimized over both camera parameters and scene points, to create a globally consistent solution in metric scale. In Fig. 1 we show an image of the Crazyflie 2.0, as well as the output from the proposed 4-point algorithm.

The main reason to use different UAVs with different components is to show the versatility and robustness of the proposed solvers, as different setups might perform differently depending on which IMU filtering technique or pre-integration method is used. Furthermore, there are cases when the distortion profile of the camera optics may not be accurately approximated by the one-parameter division model.

In the experiments we use a simple pre-integration technique [8] to obtain the estimated relative rotation. As discussed in Sec. 3, the direction of the gravity may drift, hence the 4-point methods by Ding et al. [6] will not perform optimally, as one would have to select a filtering technique that preserves the gravity direction. To remedy this situation, we use the ground truth poses to align the direction for this method every frame, and apply the estimated relative ori-
Figure 6. (Left:) The mid-sized UAV used in the dataset [41] and one of the input images from the Indoor sequence. (Right:) The output of the proposed 4-point algorithm, where the distortion parameter was estimated using histogram voting. Note that the checkerboard pattern on the floor is a quadrilateral box in real-life; however, it is significantly distorted (red) in the input image. The rectified image, on the other hand, displays a quadrilateral box (green). This indicates that lines are mapped to lines and that the pinhole camera model is applicable. This is strong evidence that the distortion profile has been accurately estimated, and that the one-parameter division model (7) is sufficient for the optics used on the UAV.

entation. Note that this gives the exact same relative error per input and frame as for the other solvers, but guarantees that the input to the competing 4-point solver is aligned with the gravity direction. Although these circumstances are not plausible in real-life using pre-integration, this assures that the result is not skewed by the choice of IMU filtering technique. The reported statistics for the 4-point method [6], however, are optimistic, as as the first input orientation is perfectly aligned in the experiments. In real-life situations they might perform worse.

Another important aspect to note is that we use consecutive frames, as we want to minimize potential drift. This means that the translation between frames might not be very long—typically not more than 30 cm, causing the baseline to be short. In [6] the authors only used frames \{1, 11\}, \{2, 12\},...\} to avoid this situation; however, we want to utilize this data, as it is important for real-life applications concerning UAV positioning, e.g. moving obstacle avoidance.

We use the following error metrics to measure the quality of the pose reconstruction

\[ e_R = \arccos \left( \frac{\text{tr}(R_{GT} R_{est}^T) - 1}{2} \right), \] (11)

\[ e_t = \arccos \left( \frac{t_{GT}^T t_{est}}{\|t_{GT}\| \|t_{est}\|} \right), \] (12)

\[ e_f = \frac{|f_{GT} - f_{est}|}{f_{GT}}, \] (13)

which have been used in a number of works [5,6,37,40,41]. The distortion profile is not as simple to measure, and we will discuss this in Sec. 7.2.

7.1. Pose Estimation with Rectified Images

In this section, we use the rectified images for the methods that do not compensate for radial distortion (the proposed 3-point solver, the 4-point solver [6] and the 6-point solver [23]), while the proposed 4-point solver is given the distorted images as input. We do not use the 7-point solver [14] in this comparison, since it is not a feasible competitor in real situations due to its computational complexity. Each method is given 1,000 RANSAC iterations with the same reprojection threshold, and no extra local optimization techniques are applied. The results are shown in Tab. 2.

From the result, we note that there is a slight advantage in terms of accuracy in favor of the method by Ding et al. [6] compared to the proposed 3-point solver for the dataset from [41] generated with the mid-sized UAV; however, the roles are reversed when looking at the new dataset with the Crazyflie 2.0 UAV. Note that the rotation error for the proposed 3-point (and 4-point, since they are identical) are generally larger for the mid-sized UAV (in the range 0.4–0.6 degrees) whereas the error input to the Crazyflie 2.0 is slightly smaller (roughly 0.2 degrees on average). This could be explained by the internal calibration between the IMU and the camera being more accurate on the Crazyflie 2.0. Furthermore, we would like to emphasize that the differ-
### Table 2. Pose estimation error for the two datasets with rectified input images to all but the proposed 4-point method. Note that the method by [6] is given the first ground truth rotation matrix as input to ensure the assumed alignment with the y-axis. Therefore, it is likely to perform worse in real-life.

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Carpet</th>
<th>Indoor</th>
<th>Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rot. Error (deg.)</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>0.435</td>
<td>2.243</td>
<td>24.804</td>
<td>0.435</td>
</tr>
<tr>
<td></td>
<td>3.77</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td></td>
<td>0.435</td>
<td>2.243</td>
<td>24.804</td>
<td>0.435</td>
</tr>
<tr>
<td></td>
<td>3.77</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trans. Error (deg.)</th>
<th>Median</th>
<th>Median</th>
<th>Median</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.75</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td></td>
<td>7.02</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td></td>
<td>7.02</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td></td>
<td>3.75</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td></td>
<td>7.02</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td></td>
<td>7.02</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td>Focal Error (perc.)</td>
<td>Median</td>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td></td>
<td>7.02</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
<tr>
<td></td>
<td>7.02</td>
<td>6.336</td>
<td>6.104</td>
<td>6.336</td>
</tr>
</tbody>
</table>

### Table 3. Pose estimation error with unrectified input images to all methods. The proposed 4-point method is the only method capable of correcting for radial distortion artifacts.

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Carpet</th>
<th>Indoor</th>
<th>Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rot. Error (deg.)</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trans. Error (deg.)</th>
<th>Median</th>
<th>Median</th>
<th>Median</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td>Focal Error (perc.)</td>
<td>Median</td>
<td>Median</td>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
<td>0.197</td>
</tr>
</tbody>
</table>

7.2. Pose Estimation with Raw Images

We now turn our attention to using distorted input images. This scenario is interesting for UAV operators who wish to change optics out in the field without intermediate calibration procedures. The same input sequences as in Sec. 7.1 are used; however, they are not rectified prior to estimating the image point correspondences. The results are shown in Tab. 3. Unsurprisingly, our 4-point method outperforms the other methods that cannot correct for distortion artifacts. What is perhaps more interesting is that the performance, in general, is better than the visual-only 6-point method [23] on rectified data. These observations suggest that the radial distortion auto-calibration approach is practically feasible using the proposed solver.

As optics, in general, are not perfectly approximated by the one-parameter division model, it is non-trivial to express the performance of the radial distortion correction. Instead, we rely on an ocular inspection of the estimated radial distortion parameter for two sequences. In Fig. 1 and Fig. 6 we show the rectifications, using the estimated radial distortion parameter obtained from histogram voting of the respective sequence. In the latter case, we get a clear visual confirmation of the successful estimation of the radial distortion parameter, in the form of a quadrilateral checkerboard pattern visible on the floor.

8. Conclusions

In this paper, we have investigated an assumption of ignoring the IMU drift for short time intervals. We showed that modern pre-integration methods perform well and that the relative pose problem can be solved accurately and satisfactorily using this assumption. What is most important, is that the resulting equations are significantly easier to solving, opening up the possibility to tackle problems that were previously considered extremely hard and not suitable for real-time applications. We proposed the first-ever minimal solver for simultaneously estimating the focal length, distortion profile, and motion parameters while incorporating the IMU data. Furthermore, we showed a speed-up of 800× compared to the current state-of-the-art for the partially calibrated case with unknown and equal focal length, with little to no loss in accuracy. The methods have been thoroughly tested on different UAVs with different components, in several challenging indoor and outdoor environments, demonstrating excellent performance.
References


Aleka McAdams, A. Selle, Rasmus Tamstorf, J. Teran, and Efthchos Sifakis. Computing the singular value decomposition of 3x3 matrices with minimal branching and elementary floating point operations. Technical Report 1690, University of Wisconsin, 2011. 4


Changchang Wu. P3.5p: Pose estimation with unknown focal length. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2