Minimal Solvers for Indoor UAV Positioning

Marcus Valtonen Örnhag[1], Patrik Persson[1], Mårten Wadenbäck[2], Kalle Åström[1] and Anders Heyden[1]
[1]Lund University, Sweden [2]Linköping University, Sweden

Overview

In this paper we consider a collection of relative pose problems which arise naturally in applications for visual indoor navigation using unmanned aerial vehicles (UAVs). We focus on cases where additional information from an onboard IMU is available and thus provides a partial extrinsic calibration through the gravitational vector. The solvers are designed for a partially calibrated camera, for a variety of realistic indoor scenarios, which makes it possible to navigate using images of the ground floor. Current state-of-the-art solvers use more general assumptions, such as using arbitrary planar structures; however, these solvers do not yield adequate reconstructions for real scenes, nor do they perform fast enough to be incorporated in real-time systems.

We show that the proposed solvers enjoy better numerical stability, are faster, and require fewer point correspondences, compared to state-of-the-art approaches. These properties are vital components for robust navigation in real-time systems, and we demonstrate on both synthetic and real data that our method outperforms other solvers, and yields superior motion estimation

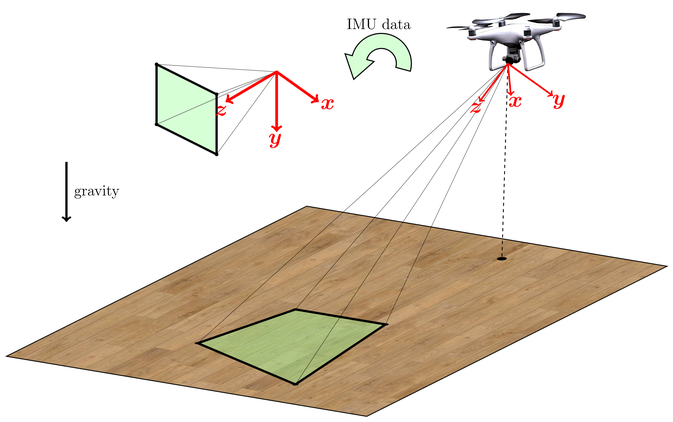

Fig. 1: The pitch and roll angles can be estimated from the IMU data, leaving the unknown yaw angle to be determined.

Our main contributions are fourfold:

1. We incorporate the known IMU data for positioning using the ground floor and a partially calibrated camera with unknown focal length,
2. The solvers are orders of magnitude faster than the current state-of-the-art,
3. We demonstrate through numerical experiments that our solvers are better than or on par with existing solvers with respect to accuracy and runtime, and
4. We run the solvers in real-time on a UAV system, demonstrating that the derived solvers are feasible for practical real-world situations.

Incorporating the IMU data

We consider the case of a common reference direction, assumed to be aligned with the gravitational direction. After a suitable change of coordinates, we may assume that

$$\boldsymbol{H}_y \sim \boldsymbol{R}_y + \boldsymbol{t}\boldsymbol{n}^T,$$

where $\boldsymbol{R}_y$ is a rotation about the y-axis (gravitational direction). Hence, the DLT equations can be written as

$$\boldsymbol{R}_2^T \boldsymbol{K}_2^{-1} \boldsymbol{x}_2 \sim \boldsymbol{H}_y \boldsymbol{R}_1^T \boldsymbol{K}_1^{-1} \boldsymbol{x}_1 \, .$$

The known matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are given by the IMU data, and consist of the pitch and roll angles for the first and second camera positions. Furthermore, we will assume that $\boldsymbol{n} = (0,\, 0,\, 1)^T$, i.e. that we navigate using the ground floor.

Noise sensitivity

The rotation error and translation error for synthethic instances are shown below

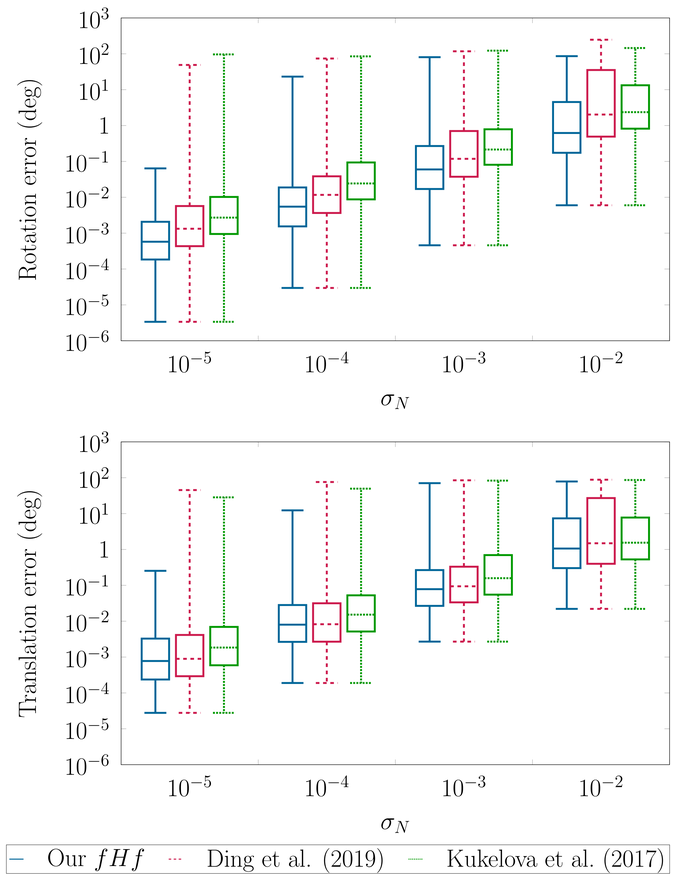

Fig. 2: Noise sensitivity comparison for Gaussian noise with standard deviation σ_N. For each noise level 1,000 random problem instances were generated.

Real experiments

The real data was captured using a monochrome global shutter camera (OV9281) with resolution 480×640 and an inertial measurement unit (MPU-9250). The extracted feature locations were undistorted to remove fish-eye effects. Ground truth and pair-wise feature matches were generated by a simultaneous localization and mapping system, where both the re-projection and IMU error were minimized; this is in order to create a globally consistent solution in metric scale. No assumptions about the scene structure were made, which lead to some matches not belonging to the ground plane, resulting in natural outliers, for the proposed method.

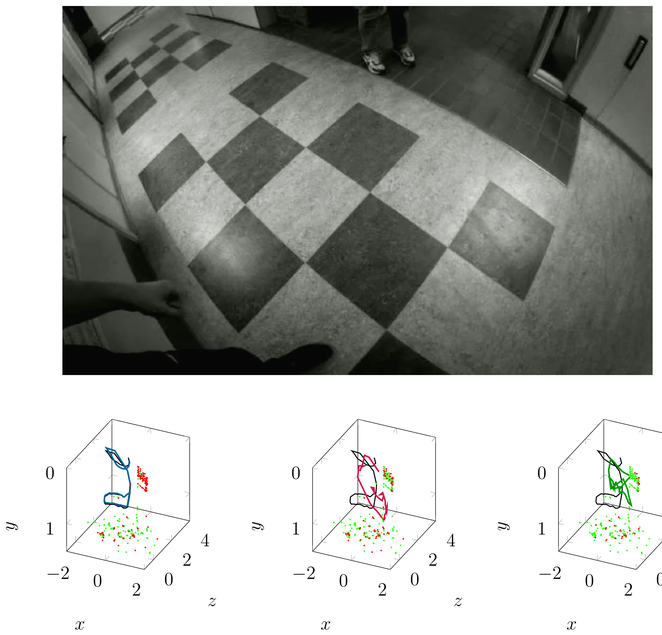

Fig. 3: Drone experiment ground truth trajectories (black) for the *indoor* sequence, and the estimated trajectories for the different solvers. Green dots indicate inliers (from at least one frame) and red dots denote outliers (all frames). *Left to right:* Our, Ding et al. (2019) and Kukelova et al. (2017). Top image: An example input prior to rectifying the image.

Table 1: Mean execution time (C++).

Case	Author	Execution time (μs)
fHf	Our	14
	Ding et al. (2019)	1052
fEf	Kukelova et al. (2017)	103
Hf	Our	5
	Ding et al. (2019)	124
Ef	Kukelova et al. (2017)	25
f_1Hf_2	Our	9
	Ding et al.. (2019)	47
f_1Ef_2	Bougnoux (1998)	27

International Conference on Pattern Recognition, 2020