

# RaD-VIO: Rangefinder-aided Downward Visual-Inertial Odometry

Bo Fu, Kumar Shaurya Shankar, Nathan Michael

**Abstract**—State-of-the-art forward facing monocular visual-inertial odometry algorithms are often brittle in practice, especially whilst dealing with initialisation and motion in directions that render the state unobservable. In such cases having a reliable complementary odometry algorithm enables robust and resilient flight. Using the common local planarity assumption, we present a fast, dense, and direct frame-to-frame visual-inertial odometry algorithm for downward facing cameras that minimises a joint cost function involving a homography based photometric cost and an IMU regularisation term. Via extensive evaluation in a variety of scenarios we demonstrate superior performance than existing state-of-the-art downward facing odometry algorithms for Micro Aerial Vehicles (MAVs).

## I. INTRODUCTION AND RELATED WORK

Recent advances in optimisation based monocular visual-inertial SLAM algorithms for MAVs have made great strides in being accurate and efficient [1]. However, in practice, these algorithms suffer from three main failure modalities - sensitivity to initialisation, undergoing motion that renders the state unobservable, and, to a lesser extent, inability to handle outliers within the optimisation. The first arises from the need for translation to accurately triangulate feature landmarks and being able to excite all axes of the accelerometer to determine scale. The second is a fundamental limit of the sensor characteristics, robot motion, and the environment, most often caused by motion in the camera direction and an absence of texture information. The third is often an artefact of sliding windows necessitated by the constraints imposed by limited compute on aerial platforms.

We believe that in order to have resilient closed loop flight it is imperative to have complementary sources of odometry. Towards this, we present an algorithm that computes metric velocity without depending on triangulation or feature initialisation, utilises observability in an orthogonal direction to a conventional forward facing camera, and is purely a frame-to-frame method. This enables it to be fast and reliable while still being accurate.

In this paper, we pursue the problem of estimating the linear and angular velocity and orientation of a micro aerial vehicle (MAV) equipped with a downward facing camera, an IMU, and a single beam laser rangefinder which measures the height of the vehicle relative to the ground.

A common strategy for performing visual odometry using downward facing cameras involves exploiting epipolar geometry using loosely-coupled [2], [3] or tightly-coupled visual-inertial methods [4], [5]. An alternate class of approaches make a planar ground assumption which enables optical

Authors are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213, USA {bofu, kshaurya, nmichael}@cmu.edu

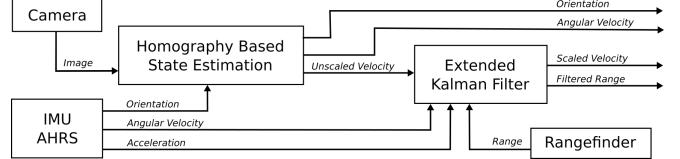


Fig. 1: System Diagram. The homography based optimisation takes in images and differential rotation estimates from the Attitude and Heading Reference System (AHRS) on the IMU and outputs an orientation and unscaled velocity, which is then used in an Extended Kalman filter to provide scaled metric velocity.

flow based velocity estimation where the camera ego motion is compensated using angular rate data obtained from a gyroscope and metric scaling obtained using an altitude sensor [6]. An issue with all such epipolar constraint based methods is that their performance is predicated on there being detectable motion between camera frames. Common failure modes for this class of techniques are situations when the camera is nearly static in hovering conditions or when it moves vertically.

These failure modes can be mitigated by explicitly encoding the planar ground assumption using the homography constraint between successive camera views. Implicit means of utilising this constraint have been presented earlier in appearance based localisation [7] where cameras are localised against a library of previously acquired images. Most relevant to our approach, the authors in [8], [9], [10] first estimate the optical flow between features in consecutive frames and then explicitly use the homography constraint and the angular velocity and ground plane orientation obtained from an inertial sensor to obtain unscaled velocity. They finally use an extended Kalman filter (EKF) to fuse the data and output metric velocity. We use this work as our baseline.

In this work, instead of using sparse visual features that are highly dependent on textured environments, we utilise a dense, direct method that makes use of all the visual information present in the camera image and couple it with angular constraints provided by an IMU within a least squares optimisation. We then fuse the result of this optimisation with altitude data from a rangefinder to obtain metric velocity.

Contributions of this work include:

- A homography based frame-to-frame velocity estimation algorithm, that is accurate and robust in a wide variety of scenes;
- An EKF structure to incorporate this with a single beam laser rangefinder signal and estimate IMU bias; and
- Extensive evaluation on a wide variety of environments with comparisons with state of the art algorithms.

## II. ESTIMATION THEORY

In this section we present the homography constraint, our optimisation strategy, and the framework to incorporate the corresponding cost functions.

### A. Homography Constraint and Parameterisation

When looking at points lying on the same plane, their projected pixel coordinates in two images ( $\mathbf{X}$  and  $\mathbf{X}'$  respectively) taken by a downward camera can be related by

$$\mathbf{X} \equiv \mathbf{H}\mathbf{X}' \quad (1)$$

where

$$\mathbf{H} = \mathbf{K}(\mathbf{R} + \mathbf{t}_0/d \cdot \mathbf{n}^T)\mathbf{K}^{-1} = \mathbf{K}(\mathbf{R} + \mathbf{t} \cdot \mathbf{n}^T)\mathbf{K}^{-1} \quad (2)$$

where  $\mathbf{X} = [x, y, 1]^T$  and  $\mathbf{X}' = [x', y', 1]^T$  are the pixel locations in previous and current image respectively,  $\mathbf{H}$  is the warp matrix,  $\mathbf{R}$ ,  $\mathbf{t}_0$  are the rotation matrix and translation vector from the second camera frame to the previous frame,  $\mathbf{t}$  is the unscaled translation,  $\mathbf{n}$ ,  $d$  are the unit normal vector and distance to the ground plane in second camera frame, and  $\mathbf{K}$  is the camera intrinsic matrix (assumed known).

During optimisation we parameterise  $\mathbf{R}$  as a Rodrigues vector  $\mathbf{r} = [r_x, r_y, r_z]^T$  and  $\mathbf{n}$  as [11]

$$\theta = \tan^{-1}(n_y/n_x) \quad (3)$$

$$\phi = \sin^{-1}(n_z) \quad (4)$$

Since the IMU provides reliable orientation information, especially for pitch and roll, out of the three possible parameterisations :  $\mathbf{p} = [t_x, t_y, t_z]^T$ ,  $\mathbf{p} = [t_x, t_y, t_z, r_x, r_y, r_z]^T$  and  $\mathbf{p} = [t_x, t_y, t_z, r_x, r_y, r_z, \theta, \phi]^T$ , we choose the second since it provides the most accurate homography optimisation and tracking performance. The underlying assumption for fixing  $\mathbf{n}$  is that the ground is horizontal and therefore the normal vector depends only on the MAV's orientation. The validity of this assumption will be evaluated in Sec. IV.

### B. Homography Estimation Cost Function

The parameters of the warp matrix  $\mathbf{H}$  are estimated by minimising the Sum of Squared Differences (SSD) error between image pixel intensities of the reference and warped images. However, a purely photometric cost minimisation may provide incorrect camera pose estimates due to a lack of observability or in the event of non-planar objects in the camera field of view. Since the IMU provides reliable orientation information, we add a penalty term which biases the homography solution and avoids these local minima.

Suppose  $\mathbf{X} = T(\mathbf{X}'; \mathbf{p})$  stands for the homography mapping parameterised by the vector  $\mathbf{p}$ , we have

$$\begin{aligned} \mathbf{p} &= \underset{\mathbf{p}}{\operatorname{argmin}} (f_{photo} + f_{imu}) \\ f_{photo} &= \sum_{j=1}^N \|I(T(\mathbf{X}'_j; \mathbf{p})) - I'(\mathbf{X}'_j)\|^2 \\ f_{imu} &= (\mathbf{p} - \mathbf{p}_0)^T \mathbf{W}(\mathbf{p} - \mathbf{p}_0) \end{aligned} \quad (5)$$

where  $\mathbf{p}_0$  is the initial guess obtained from IMU,  $\mathbf{W}$  is a diagonal penalty weight matrix,  $I$  and  $I'$  are the previous

and current image respectively, and  $\mathbf{X}'_j$  is a pixel position in the evaluation region of current image.

### C. Gauss-Newton Optimisation

We solve for the optimal parameters using iterative Gauss-Newton optimisation. After concatenating all the intensity values of pixel  $X'_j$  in a vector, the Taylor expansion is

$$\begin{aligned} f(\mathbf{p} + \Delta\mathbf{p}) &= \|\mathbf{i}(\mathbf{p}) + \mathbf{G}\Delta\mathbf{p} - \mathbf{i}'\|^2 + \\ &\quad (\mathbf{p} + \Delta\mathbf{p} - \mathbf{p}_0)^T \mathbf{W}(\mathbf{p} + \Delta\mathbf{p} - \mathbf{p}_0) \end{aligned} \quad (6)$$

where  $\mathbf{i}(\mathbf{p}) = [I(T(\mathbf{X}'_1; \mathbf{p})), \dots, I(T(\mathbf{X}'_N; \mathbf{p}))]^T$  and  $\mathbf{i}' = [I'(\mathbf{X}'_1), \dots, I'(\mathbf{X}'_N)]^T$ . The iterative update to the parameter vector ends up being

$$\Delta\mathbf{p} = (\mathbf{G}^T \mathbf{G} + \mathbf{W})^{-1} (\mathbf{G}^T (\mathbf{i}' - \mathbf{i}(\mathbf{p})) + \mathbf{W}(\mathbf{p}_0 - \mathbf{p})) \quad (7)$$

where  $G$  is the Jacobian of the photometric residual term. Note that as an implementation optimisation we only choose pixels with a high gradient magnitude similar to [12]. This significantly speeds up computation of the update with negligible loss in accuracy. The detailed timing performance is discussed in Sec. IV.

## III. VISUAL-INERTIAL FUSION

The optimisation in the previous section outputs an unscaled translation. Inspired by [9], we use an EKF to scale it to metric and additionally filter the frame-to-frame noise.

### A. Definition

In the following section the superscripts and subscripts  $C$  and  $I$  imply a quantity in the camera and IMU frames respectively. The state vector contains camera velocity in the camera frame  ${}^C\mathbf{v}$ , distance to the plane from the camera  $d$ , and the linear acceleration bias in the IMU frame  ${}^I\mathbf{b}$ .

$$\mathbf{x} = [{}^C\mathbf{v}^T, d, {}^I\mathbf{b}^T]^T, \quad {}^C\mathbf{v}, {}^I\mathbf{b} \in \mathbb{R}^3, d \in \mathbb{R}$$

### B. Prediction

The derivative of  ${}^C\mathbf{v}$  can be modeled [9] as

$$\begin{aligned} {}^C\dot{\mathbf{v}} &= {}^C\mathbf{R}_I ({}^I\mathbf{a} + [{}^I\dot{\omega}]_\times {}^I\mathbf{p}_{IC} + [{}^I\omega_m]_\times^2 {}^I\mathbf{p}_{IC}) - [{}^C\omega_m]_\times {}^C\mathbf{v} \\ &\approx {}^C\mathbf{R}_I ({}^I\mathbf{f}_m + {}^I\mathbf{g} + [{}^I\omega_m]_\times^2 {}^I\mathbf{p}_{IC}) - [{}^C\omega_m]_\times {}^C\mathbf{v} \end{aligned} \quad (8)$$

where  ${}^C\mathbf{R}_I$  is the rotation matrix from IMU frame to camera frame,  ${}^I\mathbf{a}$  and  ${}^I\mathbf{g}$  are the acceleration and gravity in the IMU frame,  ${}^I\mathbf{f}_m$  and  ${}^I\omega_m$  are the raw linear acceleration and angular velocity measured by the IMU (subscript  $m$  denotes raw measurement from visual odometry, IMU, or range finder), and  ${}^C\omega$  is the angular velocity in the camera frame. The subscript  $\times$  denotes the skew symmetric matrix of the vector inside the bracket.

Therefore, the prediction process in discrete EKF can be written as

$${}^C\hat{\mathbf{v}}[k]_{k-1} = {}^C\hat{\mathbf{v}}[k-1] + \tau {}^C\dot{\mathbf{v}}[k] \quad (9)$$

$$\hat{d}[k]_{k-1} = \hat{d}[k-1] + \tau {}^C\hat{\mathbf{v}}[k-1]^T {}^C\mathbf{n}[k] \quad (10)$$

$${}^I\hat{\mathbf{b}}[k]_{k-1} = {}^I\hat{\mathbf{b}}[k-1] \quad (11)$$

where  $\tau$  is the time step and  ${}^C\mathbf{n}$  is the normal vector in camera frame (an alias for  $\mathbf{n}$  defined in Sec. II-A). The predicted states are denoted using  $\hat{\mathbf{x}}[k]_{k-1}$ . Using the Jacobian matrix  $\mathbf{G}[k]_{k-1}$  the predicted covariance matrix of system uncertainty  $\Sigma[k]_{k-1} \in \mathbb{R}^{7 \times 7}$  is updated as

$$\Sigma[k]_{k-1} = \mathbf{G}\Sigma[k]\mathbf{G}^T + \mathbf{V} \begin{bmatrix} \text{cov}({}^I\mathbf{f}_m) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \text{cov}({}^I\omega_m) \end{bmatrix} \mathbf{V}^T \quad (12)$$

where

$$\begin{aligned} \mathbf{G} &= \frac{\partial \hat{\mathbf{x}}[k]_{k-1}}{\partial \hat{\mathbf{x}}[k-1]} \\ &= \begin{bmatrix} \mathbf{I}_3 - \tau [{}^C\omega]_x & \mathbf{0}_{3 \times 1} & -\tau {}^C\mathbf{R}_I \\ \tau {}^C\mathbf{n}^T & 1 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{I}_3 \end{bmatrix} \in \mathbb{R}^{7 \times 7} \end{aligned} \quad (13)$$

$$\begin{aligned} \mathbf{V} &= \begin{bmatrix} \frac{\partial \hat{\mathbf{x}}[k]_{k-1}}{\partial {}^I\mathbf{f}_m} & \frac{\partial \hat{\mathbf{x}}[k]_{k-1}}{\partial {}^I\omega_m} \end{bmatrix} \\ &= \begin{bmatrix} \tau {}^C\mathbf{R}_I & \tau ({}^C\mathbf{R}_I \mathbf{M} + [{}^C\mathbf{v}]_x {}^C\mathbf{R}_I) \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \in \mathbb{R}^{7 \times 6} \end{aligned} \quad (14)$$

$$\mathbf{M} = ({}^I\omega_m^T {}^I\mathbf{p}_{IC}) \mathbf{I}_3 + {}^I\omega_m {}^I\mathbf{p}_{IC}^T - 2 {}^I\mathbf{p}_{IC} {}^I\omega_m^T \quad (15)$$

### C. Update

When both unscaled translation between two frames  $\mathbf{t}_m[k]$  and range sensor signal  $l_m[k]$  are available for update, the measurement vector  $\mathbf{z}_m[k]$  is

$$\mathbf{z}_m[k] = \begin{bmatrix} \mathbf{t}_m[k]/\tau \\ l_m[k] n_{z_m}[k] \end{bmatrix} \quad (16)$$

where the  $n_{z_m}[k]$  is the z component of  ${}^C\mathbf{n}$ , and the subscript  $m$  denotes direct measurements. The predicted measurement based on  $\hat{\mathbf{x}}[k]_{k-1}$  is

$$\hat{\mathbf{z}}[k]_{k-1} = \begin{bmatrix} {}^C\hat{\mathbf{v}}([k]_{k-1})/\hat{d}([k]_{k-1}) \\ \hat{d}([k]_{k-1}) \end{bmatrix} \quad (17)$$

Calculating the Kalman gain  $\mathbf{K}[k] \in \mathbb{R}^{7 \times 4}$

$$\mathbf{K}[k] = \Sigma J^T (\mathbf{J}\Sigma\mathbf{J}^T + \text{cov}(\mathbf{z}_m))^{-1} \quad (18)$$

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} \hat{\mathbf{z}}[k]_{k-1} & \hat{\mathbf{z}}[k]_{k-1} & \hat{\mathbf{z}}[k]_{k-1} \\ {}^C\hat{\mathbf{v}}[k]_{k-1} & \hat{d}[k]_{k-1} & {}^I\hat{\mathbf{b}}[k]_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\hat{d}[k]_{k-1}} \mathbf{I}_3 & \frac{-{}^C\hat{\mathbf{v}}[k]_{k-1}}{\hat{d}[k]_{k-1}^2} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{1 \times 3} & 1 & \mathbf{0}_{1 \times 3} \end{bmatrix} \in \mathbb{R}^{4 \times 7} \end{aligned} \quad (19)$$

Estimates  $\hat{\mathbf{x}}[k]$  and  $\Sigma[k]$  are updated accordingly as

$$\hat{\mathbf{x}}[k] = \hat{\mathbf{x}}[k]_{k-1} + \mathbf{K}[k] (\mathbf{z}_m[k] - \hat{\mathbf{z}}[k]_{k-1}) \quad (20)$$

$$\Sigma[k] = (\mathbf{I}_7 - \mathbf{K}\mathbf{J}) \Sigma[k]_{k-1} \quad (21)$$

## IV. EVALUATION

We evaluate the performance of our approach on a wide variety of scenarios and compare and contrast performance with state of the art algorithms. We first present experimental setup and results in simulation followed by those with real-world data obtained from an aerial platform.

### A. Benchmarks and Metrics

Our method (RaD-VIO) is compared to the tracker proposed in [10] (Baseline), for which we implement the optical flow method described in [8] and, for fair comparison, use the same EKF fusion methodology as our approach. The resulting tracking errors of the EKF with a range finder are much smaller than those when using the EKF in [9]. We choose this as the baseline since it is also based on the homography constraint and assumes local planarity. Additionally, we also compare with a state-of-the-art monocular visual-inertial tracker VINS-Mono [5] without loop closure (VINS-D, VINS-downward).

The metrics used are Relative Pose Error (RPE) (the interval is set to 1s) and Absolute Trajectory Error (ATE) [13]. For ATE, we only report results in the xy plane since the altitude is directly observed by the rangefinder. We also report the number of times frame-to-frame tracking fails in Fig. 4. Since RaD-VIO and Baseline output velocity, the position is calculated using dead-reckoning. Since a lot of our trajectories are not closed loops, instead of reporting ATE we divide it by the cumulative length of the trajectory (computed at 1s intervals) in the horizontal plane to get the relative ATE. We try to incorporate an initial linear movement in test cases to initialise VINS-D well, but a good initialisation is not guaranteed. For error calculation of VINS-D, we only consider the output of the tracker after it finishes initialisation.

### B. Simulation Experiments

We utilise AirSim [14], a photorealistic simulator for aerial vehicles for the purpose of evaluation. The camera generates  $240 \times 320$  images at a frame rate of 80 Hz. The focal length is set to 300 pixels. The IMU and the single beam laser rangefinder output data at 200 Hz and 80 Hz respectively, and no noises are added.

1) *Simulation Test Cases*: For the tracking system to work, the following assumptions or conditions should be met or partly met:

- The ground should be planar (homography constraint)
- It should be horizontal (parameterisation choice)
- The scene should be static (constancy of brightness)
- Small inter-frame displacement (applicability of warp).

Therefore, the following ten test scenarios are designed to evaluate the algorithms:

- **p1**: All assumptions met - ideal conditions
- **p2**: Viewing planar ground with low texture
- **p3**: Viewing planar ground with almost no texture
- **p4**: Viewing planar ground with moving features
- **p5**: Vehicle undergoing extreme motion
- **p6**: Camera operating at a low frame rate
- **s1**: Viewing a sloped or a curved surface
- **m1**: Viewing a plane with small clutter
- **m2**: Viewing moving features with small clutter
- **c1**: Viewing a plane with large amounts of clutter

Each of these cases are tested in diverse environments including indoors, road, and woods. We use 42 data sets in

total for evaluation. Fig. 2 shows some typical images from these datasets.

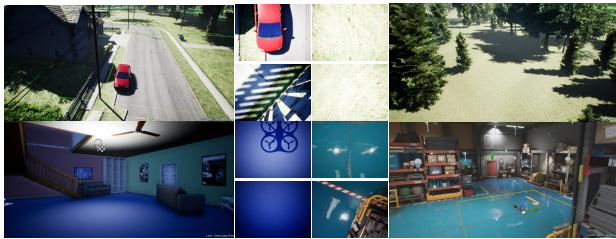


Fig. 2: Sample views of environments used in the simulation experiments defined in Sec. IV-B.1

*2) Simulation Results and Discussion:* The RPE and relative ATE of all test cases are shown in Fig. 3. For relative ATE our method outperforms both Baseline and VINS-D in almost all the test cases. The velocity error in Fig. 3 (c) contains both velocity bias and random error, and compared to VINS-D our method generates similar velocity errors when the planar assumption is satisfied, but larger when it is not. This error is much smaller than Baseline.

Sometimes, VINS-D takes a lot of time to initialise the system, and doesn't do well right after initialisation. During the test, VINS-D occasionally outputs extremely wrong tracking results, and it has difficulty in initialising/reinitialising through three test cases (challenging lighting condition, too few texture, or extreme motion). In contrast our tracker never generates any extreme results due to implicitly being constrained by the frame-to-frame displacement.

The simulation shows that RaD-VIO is overall more accurate than compared to Baseline, and when all the assumptions are met, it is slightly more accurate than VINS-D. The test cases demonstrate that the proposed tracker is robust to a lot of non-ideal conditions.

### C. Real-world Experiments

*1) Setup and Test Cases:* To verify the real-world performance of our proposed approach, indoor and outdoor data was collected. The indoor data was obtained from a custom hexrotor platform in a motion capture arena. We use a MatrixVision BlueFox camera configured to output  $376 \times 240$  resolution images at a frame rate of 60 Hz, and a wide angle lens with a focal length of 158 pixels. The frame rates of the TeraRanger One rangefinder and the VN-100 IMU are 300 Hz and 200 Hz respectively. The standard deviation of IMU angular velocity and linear acceleration are 0.02 rad/s and 1 m/s<sup>2</sup> respectively. The indoor ground truth is provided by a VICON motion capture system. For outdoor data, only GPS data is provided as a reference, and we use an Intel Aero Drone<sup>1</sup> for data collection. The same algorithm parameters as in the simulation experiments were used for the vision and optimisation frontends, and the parameters for the EKF were tuned based on the sensor noise for the respective configuration.

<sup>1</sup><https://www.intel.com/content/www/us/en/products/drones/aero-ready-to-fly.html>

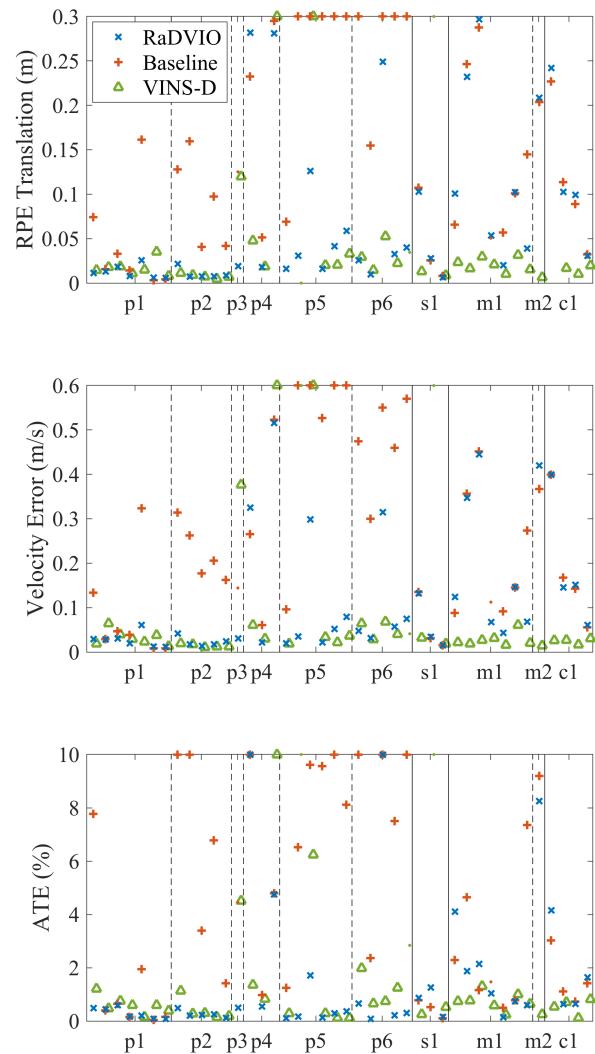


Fig. 3: Comparison of the algorithms on the simulation datasets described in Sec.IV-B.1. Solid lines separate the cases where the planar assumption is met, not met or it is a slope, and dashed lines separates other assumptions. The point marker means the data point is not reliable due to a large number of tracking failures. The errors drawn at the upper boundary are clipped.

We evaluate 10 flight data sets collected according to the same classification criteria. The severity of moving features, and the platform motion is comparatively more significant in the corresponding real data sets. Note that the abbreviation of the test cases have the same meaning as in simulation section we mentioned before.

*2) Experiment Results and Discussion:* The tracking errors are shown in Fig. 7. Similar to the simulation results for relative ATE our tracker is on average better than both Baseline and VINS-D. As for the translation part of RPE, our method is better or no worse than the other two methods except for the last case in p4 (moving features) and p5 (extreme motion). In these cases the robust sparse feature selection in VINS-D avoids being overly influenced. RaD-VIO is not as robust to extreme motion as it is in simulation, and there are two reasons: the IMU input is more noisy and

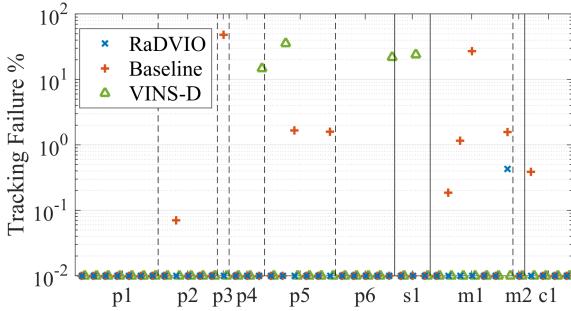


Fig. 4: Percentage of tracking failures: The failure instances of Baseline and RaD-VIO are calculated by a per-frame counter, while that of VINS-D is estimated according to gap between messages, not including time taken for initialisation.

the wider field of view ends up capturing objects outside the ground plane that adversely affect the alignment. For the rotation component of RPE, both Baseline and RaD-VIO are better than VINS-D; this is because both approaches use the IMU rotation directly.

The experiment results show that the proposed method is able to work well in real world even in the presence of sensor synchronisation issues and large noises in the IMU signal.

#### D. Timing Benchmarks

We evaluated the tracking framerate of RaD-VIO on a desktop PC with an Intel i7-6700K CPU and also onboard on an NVIDIA TX2. The frame rate of the tracker is on average 150Hz over all of our datasets on a PC and 55 Hz onboard, see Table I. Data sets with less features and large inter frame displacements result in lower frame rates due to longer time required for convergence.

Category	Mean	$\sigma$	Min	Max
Desktop	153	15.7	107	181
Onboard	55	7.0	40	65

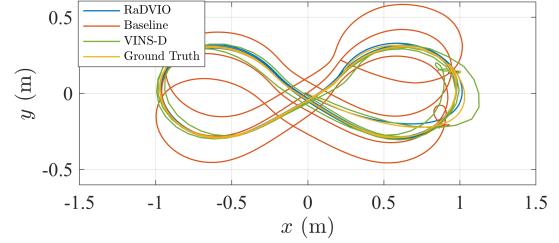
TABLE I: Frame Rate Evaluation.

## V. CONCLUSIONS AND FUTURE WORK

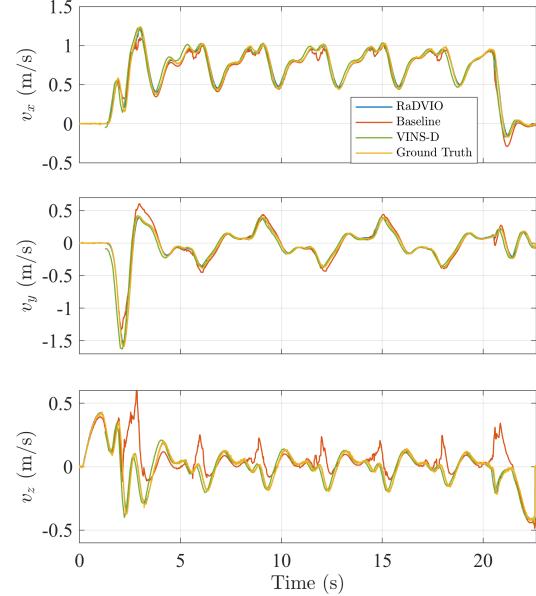
We present a framework to obtain 6 degree of freedom state estimation on MAVs using a downward camera, IMU, and a downward single beam rangefinder. The proposed approach first extracts the rotation and unscaled translation between two consecutive image frames based on a cost

Conditions	Baseline	RaD-VIO	VINS-D
Ideal (p1)	++	+++*	+++
Low texture (p2)	+	+++	+++
Negligible texture (p3)	failure	+++	+
Moving features (p4, m2)	-	-	++
Extreme motion (p5)	-	+++	+++
Low image Hz (p6)	-	+++	++
Slope (s1)	+++	+++	+++
Medium clutter (m1)	+	++	+++
High clutter (c1)	+	+	+++

TABLE II: Qualitative performance comparison on simulation data. +++:Low or no tracking failures, ++:Occasional failures, +:Frequent failures, -:Works poorly



(a) Dead-reckoned odometry tracks (aligned according to the calculation of ATE).



(b) Corresponding velocities in MAV body frame.

Fig. 5: Comparison of algorithms in simulation for a figure 8 trajectory.

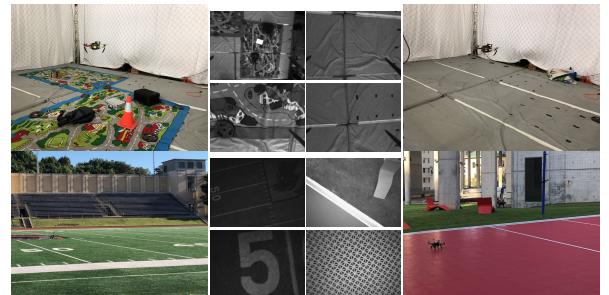


Fig. 6: Sample views of environments used in the real-world experiments discussed in Sec. IV-C.1

function that combines dense photometric homography based alignment and a rotation prior from the IMU, and then uses an EKF to perform sensor fusion and output a filtered metric linear velocity.

Extensive experiments in a wide variety of scenarios in simulations and in real-world experiments demonstrate the accuracy and robustness of the tracker under extenuating circumstances. The performance exceeds the frame-to-frame tracking framework proposed in [8], [10], and is slightly better than a current state of the art monocular visual-inertial odometry algorithm. Baseline fails in high clutter,

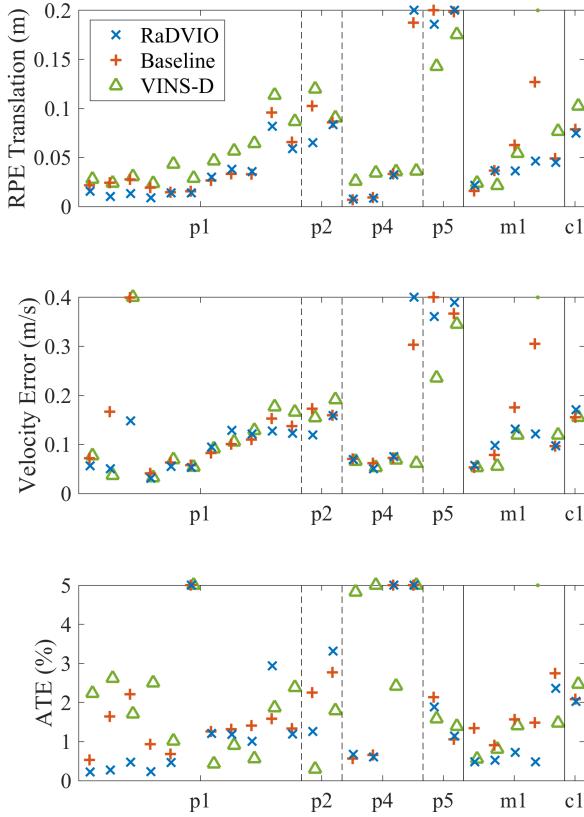


Fig. 7: Comparison of the algorithms on the real-world datasets described in Sec.IV-C.1. Solid lines separate the cases where the planar assumption is met, not met or it is a slope, and dashed lines separates other assumptions. The point marker means the data point is not reliable due to a large number of tracking failures. The errors drawn at the upper boundary are clipped.

extreme motion, or low texture, VINS-D fails when it doesn't initialise and in low texture. RaD-VIO degrades when in high clutter, but crucially it is stable and never generates extremely diverged state estimates (as triangulation based optimisation methods are susceptible to), and can operate at a high frame rate. The ability to run on high frame rate image streams ensures that consecutive images have high overlap and helps mitigate the common issue of poor performance when close to the ground. A qualitative comparison of the performance on simulation data is shown in Table II.

To relax the planar assumption in the proposed method, we replaced the SSD error between pixel intensities with Huber and Tukey loss functions [15], [16]. The accuracy improvement in cluttered environment was minor and offset by the increase in computational costs. In future work we aim to address the weakness of the planar assumption through means of explicitly accounting for it within the formulation.

Additionally, in line with our introductory statements, we intend to couple the performance of RaD-VIO with a conventional forward facing sliding window VIO algorithm to develop a resilient robotic system that exploits the individual strengths of both odometry approaches.

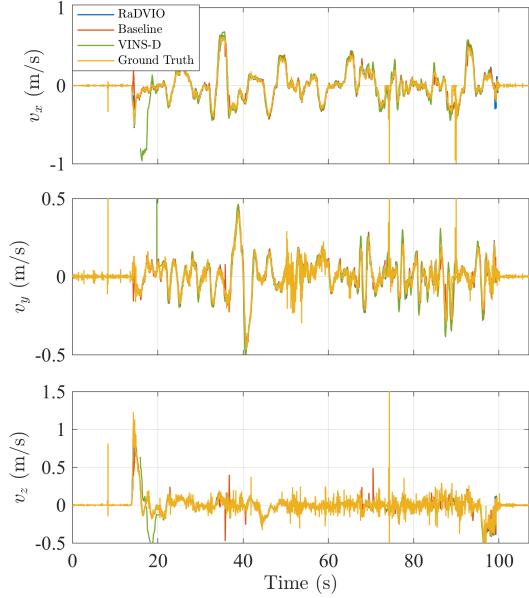


Fig. 8: Comparison of velocities in MAV body frame in a real-world indoors dataset with motion capture ground truth.

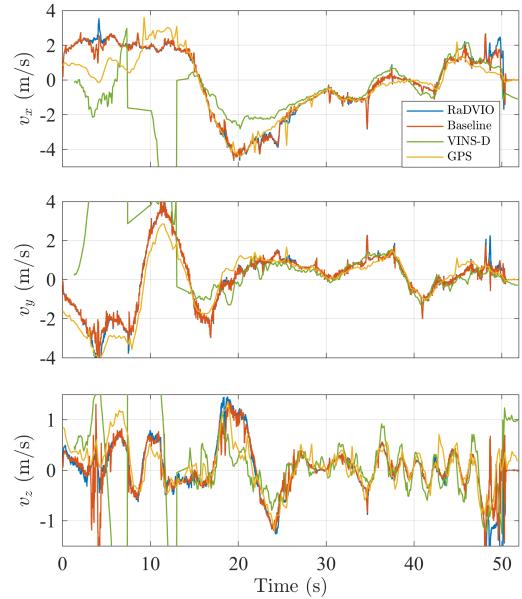


Fig. 9: Comparison of velocities in MAV body frame for an outdoors trajectory (spans a  $25m \times 45m$  region in a playground).

## REFERENCES

- [1] J. Delmerico and D. Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2018.
- [2] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, “Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments,” in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2012.
- [3] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “Svo: Semidirect visual odometry for monocular and multicamera systems,” *IEEE Transactions on Robotics*, vol. 33, no. 2, 2017.
- [4] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2007.
- [5] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, no. 99, 2018.
- [6] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, “An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications,” in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2013.
- [7] B. Steder, G. Grisetti, C. Stachniss, and W. Burgard, “Visual slam for flying vehicles,” *IEEE Transactions on Robotics*, vol. 24, no. 5, 2008.
- [8] V. Grabe, H. H. Bülfhoff, and P. R. Giordano, “On-board velocity estimation and closed-loop control of a quadrotor uav based on optical flow,” in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2012.
- [9] V. Grabe, H. H. Bülfhoff, and P. R. Giordano, “A comparison of scale estimation schemes for a quadrotor uav based on optical flow and imu measurements,” in *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2013.
- [10] V. Grabe, H. H. Bülfhoff, D. Scaramuzza, and P. R. Giordano, “Nonlinear ego-motion estimation from optical flow for online control of a quadrotor uav,” *The Intl. Journal of Robotics Research*, vol. 34, no. 8, 2015.
- [11] D. Crispell, J. Mundy, and G. Taubin, “G taubin, parallax-free registration of aerial video,” in *Proc. of the British Machine Vision Conf.*, 2008.
- [12] K. S. Shankar and N. Michael, “Robust direct visual odometry using mutual information,” in *Proc. of IEEE Intl. Symposium on Safety, Security, and Rescue Robotics*, 2016.
- [13] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, “On measuring the accuracy of slam algorithms,” *Autonomous Robots*, vol. 27, no. 4, 2009.
- [14] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, 2017.
- [15] P. J. Huber *et al.*, “Robust estimation of a location parameter,” *The annals of mathematical statistics*, vol. 35, no. 1, 1964.
- [16] R. A. Maronna, R. D. Martin, and V. J. Yohai, “Robust statistics,” 2006.