Mid-Air Range-Visual-Inertial Estimator Initialization for Micro Air Vehicles

Martin Scheiber[1], Jeff Delaune[2], Stephan Weiss[1], and Roland Brockers[2]

***Abstract*— Monocular Visual-Inertial Odometry (VIO) has become ubiquitous for navigation of autonomous Micro Air Vehicles (MAVs). Yet, state-of-the-art VIO is still very failure-prone, which can have dramatic consequences. To prevent this, VIO must be able to re-initialize in mid-air, either during a free fall or on a constant velocity trajectory after attitude control has been re-established. However, for both of these trajectories, the visual scale cannot be observed with VIO batch initializers because of the absence of acceleration change. We propose to use a small and lightweight laser-range finder (LRF) and a scene facet model to initialize vision-based navigation at the right scale under any motion condition and over any scene structure. This new range constraint is integrated into a visual-inertial bundle-adjustment initializer. We evaluate our approach in simulation, including robustness to various parameters, and demonstrate on real data how this approach can address mid-air state estimation failure in real-time.**

I. Introduction

Autonomous, safe, and robust navigation is crucial for a micro air vehicle (MAV). In-flight pose estimation must provide accurate and robust poses for flight controllers to perform ever more complex maneuvers. Many different approaches exist, ranging from multi-sensor to minimal-sensor set state estimation. Although these approaches differ, their common ground is the need for an initial state.

Especially minimum sensor suite approaches, i.e., visual-inertial odometry (VIO) algorithms, are constrained on their estimator initialization. Most state-of-the-art VIO rely on a specific scenario or motion to start their estimator correctly. However, this limits the level of MAV autonomy since the scenario or motion might be unknown when (re-)initializing. Particularly, fully-autonomous systems should be able to initialize in all airborne scenarios, which are

(a) excitation motion,
(b) constant velocity motion, including hovering (no motion), and
(c) free-fall motion.

Excitation motions are perfect for initialization, and nearly all state-of-the-art VIO algorithms rely on excitation in their initialization technique. Similarly, filter-based algorithms can cover hover or static initialization. These scenarios also refer to the most common initialization motion, especially when performing manual or velocity-control-based takeoff. Nevertheless, constant velocity and free-fall initialization can occur in mid-air deployment or mid-air recovery scenarios. However, traditional VIO frameworks cannot handle these initialization trajectories. Hence, additional environment information is needed to provide a full state visual-inertial initialization for motion (b) or (c), removing the autonomy of such approaches.

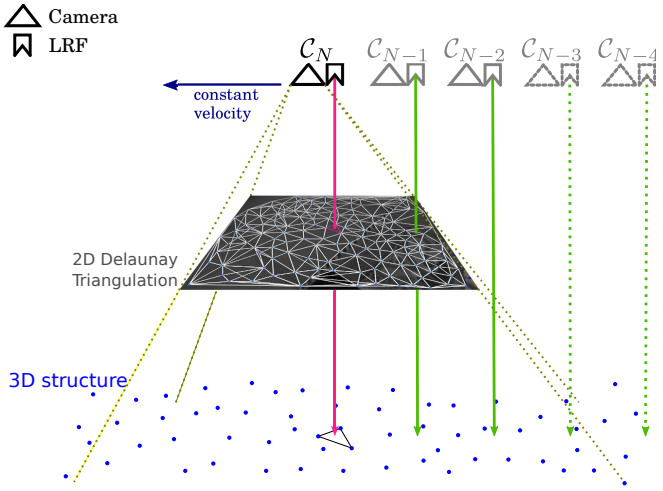

Fig. 1. Illustration of the proposed mid-air initialization algorithm for constant velocity flights. The 3D-structured scene is captured by a downward looking camera. With its generated images, features can be triangulated and divided into subgroups of triangles. Then a laser-range finder (LRF) can be used to metrically scale the Delaunay triangulated structure and camera poses in a non-linear optimization. Further, in combination with an IMU, the full MAV navigation states can be recovered in a linear way, providing a full onboard initialization. Please note that the environment is not assumed to be planar (i.e., it can be structured).

Therefore, this work aims to provide an initialization algorithm that is

- **Motion independent**: Our proposed framework can initialize in any non-zero motion, regardless of being excitation, constant velocity (as illustrated in Fig. 1), or free-fall motions.
- **Computationally fast**: Analysis of our proposed approach showed it is able to run in real-time onboard an embedded platform to provide fast initialization under time-limited motions (e.g., free-fall).
- **Free of prior knowledge**: Typically, initializers take advantage of prior knowledge, e.g., planar ground, height, level attitude, or similar. Our proposed approach works without any prior information on the motion or environment.

This work is structured as follows: Sec. II will examine state-of-the-art initialization techniques, their limitations in the in-flight reference scenarios, and why range measurement can lift these limitations. Sec. III presents our initialization

[1]These authors are with Faculty of Intelligent System Technologies, Group Control of Networked Systems, Universität Klagenfurt, Klagenfurt, Austria {martin.scheiber, stephan.weiss}@ieee.org

[2]These authors are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA {jeff.h.delaune, roland.brockers}@jpl.nasa.gov

Pre-print version, accepted Feb./2021 at ICRA21, DOI: 10.1109/ICRA48506.2021.9560913

algorithm, that can initialize in any mid-air scenario. Sec. IV takes a closer look at the influence of noisy measurements on our proposed algorithm, and Sec. V presents and discusses results on real-world constant flight experiment, as depicted in Fig. 2.

II. Related Work

A. Visual-Inertial Odometry

State-of-the-art visual-inertial state estimation frameworks comprise many different methods and algorithms. Nevertheless, such frameworks are usually grouped into two main algorithm categories: filter and optimization-based [1].

Filter approaches typically represented with a variant of extended Kalman filter (EKF) [2]. Filter based visual-inertial estimators are able to quickly propagate the state and its covariance and provide information needed for flight control using high-frequency information from the inertial measurement unit (IMU). With the IMU typically modeled as input for the system dynamics and therefore generating growing uncertainties over time, a camera sensor can provide a pose update to correct eventual drift and to decrease the uncertainty. Filter approaches shine by their ability to efficiently retain past information through marginalization implicitly in the error covariance matrix, allowing estimations without the need for time-consuming iterative optimizations. Filter-based frameworks can be divided into tightly and loosely coupled estimators [3]. Loosely coupled estimators [4], [5] perform the visual pose calculation independently from the state update and include a metric scale in their state definition. In comparison, tightly coupled systems use the include the tracked features directly in their dynamics to update and correct the state [6], [7], [8], [9], [10].

Non-linear optimization-based algorithms iteratively perform a least-square approach to converge to a state estimate [3]. The most commonly used optimization is the bundle-adjustment (BA) that minimizes the re-projection error of tracked features. The BA can be used for vison-only systems such as ORB-SLAM [11], SVO [12], or fused with inertial measurements as the *Robust and Versatile Monocular Visual-Inertial State Estimator* (VINS-Mono) [13] or *Open Keyframe-based Visual-Inertial SLAM* (OKVIS) [14] showed. Their advantage is that they can approach with sufficient iterations they can achieve better estimation quality. However, they require translation to triangulate a map. Further, they are computationally more costly since they optimize over past measurements. This problem has been mitigated in recent years as onboard processing power has increased, and through marginalization of past information.

As both nonlinear filtering and optimization approaches find the local optima, they are dependent on an accurate initialization of the state vector in the vicinity of the global optima.

B. Initialization

Robustness and performance of both filter- and optimization-based algorithms depend on the quality of the initialization routine. The former require an initial pose and velocity state estimates, which can be zero motion

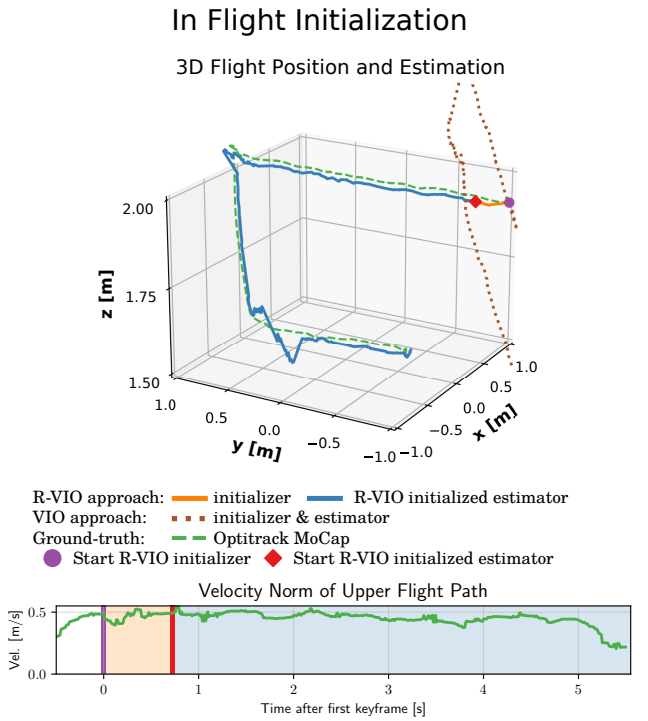

Fig. 2. Real-world experiment for initializing the proposed framework under a constant velocity flight. The initialization is triggered at (pink point), and the next 10 image frames (i.e., 0.33 s) are taken for the initialization window (orange estimates). After computing the initial navigation states (after approx. 0.75 s), the estimator VINS-Mono is initialized (red point) and continues with a visual-inertial navigation (blue estimates). The norm of the velocity throughout the initialization phase, computed with the position derivatives from the motion capturing system, are shown in the lower plot.

(assuming MAV starting on the ground before take-off). This estimate has to be relatively close to the actual value in order for the filter to converge. On the other hand, optimization-based approaches need an initial map and visual scale.

As an example of mid-air self-initialization without particular excitation motions, several studies have been presented that address the throw-and-go (TnG) problem under specific assumptions: [15] used height assumption to provide an initial estimate to their filter-based estimator, whereas [16] required an attitude estimation before the fall, flat ground surface, and horizontal translation to triangulate the initial structure and derive the metric scale for their optimization-based estimator. Further, in our previous work [17], we managed to initialize in a free-fall by aligning the magnitude of visual acceleration to the magnitude of gravity.

These free-fall initialization approaches are limited to that exact scenario and prior knowledge or assumptions and cannot be applied to horizontal motion at constant velocities. Nevertheless, IMU-pre-integration [18] can provide an opportunity to unify the mid-air self-initialization approaches in one framework and remove pre-initialization assumptions. E.g., methods with visual-inertial optimization in initialization, such as VINS-Mono, OKVIS, or OrbSLAM3 [19], rely on the IMU pre-integration to generalize their initialization algorithm to all visual-inertial observable motions.

We selected VINS-Mono as state-of-the-art algorithm to compare our approach against because of both maturity

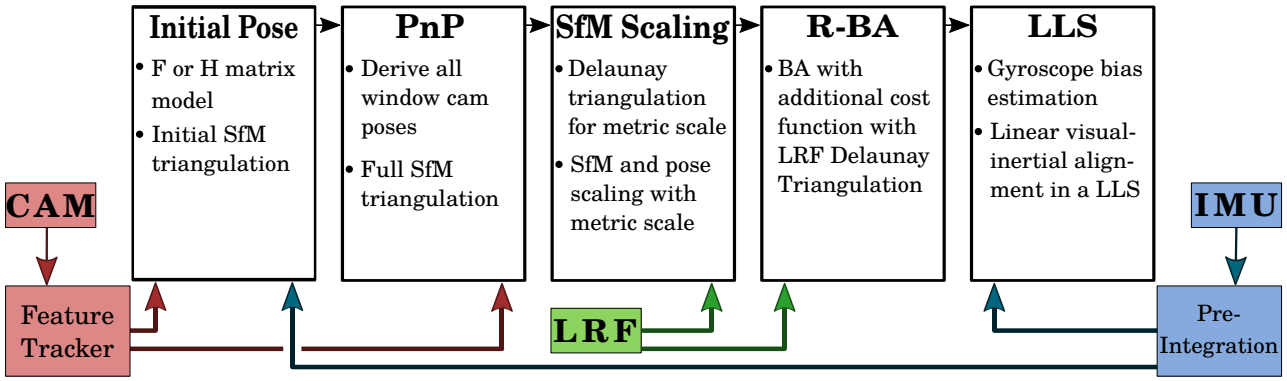

Fig. 3. The proposed Range-Visual-Inertial initialization framework. Images are used to derive the initial camera poses using the Fundamental or Homography matrix method (Sec. III-A). First the scene structure from motion (SfM) is derived using a perspective-n-point (PnP) approach (Sec. III-B). Second, this structure is scaled metrically with the range measurements received by the LRF (Sec. III-C). Then, to reduce the impact of measurement noise a range-visual bundle-adjustment (R-BA) is performed (Sec. III-D). Finally, the range-visual poses are aligned with the pre-integrated IMU measurements, to derive the globally aligned states (Sec. III-E).

and repeated good initialization performance in different scenarios. Taking a closer look on VINS-Mono's four step initialization algorithm [20], this approach first estimates the initial pose and structure using camera trigonometry, given a initialization window of N keyframes. Then a perspective-n-point (PnP) is performed to derive all other keyframe camera poses in the window and triangulate all remaining matches to form a complete structure. This structure and camera poses are then used in a visual BA to minimize measurement noise and triangulation errors, and improve the estimated poses. Given the first keyframe set as visual camera coordinate frame $\mathcal{C}$, and given the body (or IMU) coordinate frames $k = \mathcal{B}_k$ for each image at time t_k, all initialization window position and rotations, ${}^{\mathcal{C}}\mathbf{p}_k$ and ${}^{\mathcal{C}}\mathbf{R}_k$, are derived in the BA. At the last step, VINS-Mono performs a linear least-square (LLS) to linearly align these visual with the inertial IMU measurements. The latter are pre-integrated to derive the frame-to-frame position and velocity, ${}^{k}\hat{\boldsymbol{\alpha}}_{k+1}$ and ${}^{k}\hat{\boldsymbol{\beta}}_{k+1}$, respectively. Equ. (1) describes the LLS that solves for the remaining state vector ${}^{k}\hat{\mathbf{x}}_{k+N} = \left[{}^{k}\hat{\mathbf{v}}_{k+1}^{\mathsf{T}}, \quad \dots, \quad {}^{k+N-1}\hat{\mathbf{v}}_{k+N}^{\mathsf{T}}, \quad {}^{\mathcal{C}}\hat{\mathbf{g}}^{\mathsf{T}}, \quad \lambda\right]^{\mathsf{T}}$ containing the camera velocities expressed in the body frame, gravity vector expressed in the initial camera frame ${}^{\mathcal{C}}\hat{\mathbf{g}}$, and metric scale λ. Further, δt_k is the frame-to-frame time difference, $\Delta^{\mathcal{C}}\mathbf{p}_k = {}^{\mathcal{C}}\mathbf{p}_{k+1} - {}^{\mathcal{C}}\mathbf{p}_k$ the frame-to-frame position difference from the BA, and ${}^{k}\mathbf{R}_{k+1}$ the body frame-to-frame rotation derived from IMU pre-integration.

$$ {}^{k}\hat{\mathbf{x}}_{k+N} = \left({}^{k}\mathbf{H}_{k+N}^{\mathsf{T}}\,{}^{k}\mathbf{H}_{k+N}\right)^{-1} \cdot {}^{k}\mathbf{H}_{k+N}^{\mathsf{T}} \cdot {}^{k}\mathbf{z}_{k+N} \tag{1} $$

with the frame-to-frame measurement matrix and vector

$$ {}^{k}\mathbf{z}_{k+1} = \begin{bmatrix} {}^{k}\hat{\boldsymbol{\alpha}}_{k+1} - {}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} + {}^{k}\mathbf{R}_{k+1}\,{}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} \\ {}^{k}\hat{\boldsymbol{\beta}}_{k+1} \end{bmatrix} \tag{2} $$

$$ {}^{k}\mathbf{H}_{k+1} = \begin{bmatrix} -\mathbf{I}_3\,\delta t_k & \mathbf{0}_3 & \frac{1}{2}\,{}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k^2 & {}^{k}\mathbf{R}_{\mathcal{C}}\,\Delta^{\mathcal{C}}\mathbf{p}_k \\ -\mathbf{I}_3 & {}^{k}\mathbf{R}_{k+1} & {}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k & \mathbf{0}_3 \end{bmatrix} \tag{3} $$

However, this final step already shows the sensor limitations of this visual-inertial algorithm using a IMU pre-integration and visual optimization method. First, one can show [21] that under constant velocity motions, the Grammian of the measurement matrix ${}^{k}\mathbf{H}_{k+N}$ is 0. Hence the matrix ${}^{k}\mathbf{H}_{k+N}^{\mathsf{T}}\,{}^{k}\mathbf{H}_{k+N}$ is singular and the LLS not solvable [22]. Similarly, in a free-fall motion, this linear formulation yields to the measurement vector ${}^{k}\mathbf{z}_{k+N}$ being $\mathbf{0}$. As a result, the estimation of the LLS Equ. (1) can only yield a state estimate of ${}^{k}\hat{\mathbf{x}}_{k+N} = \mathbf{0}$, which differs from the ground truth. Hence, in our work's two given reference scenarios, the visual-inertial approach cannot yield a correct initialization. This also corresponds to previous work performed on visual-inertial closed-form solution [23] and visual-inertial navigation system (VINS) [24] unobservability analysis. For this reason, and to the best of our knowledge, there are no previous works attempting to initialize a VINS system in a constant velocity flight. Therefore, in the next section, we will present a range-visual-inertial approach that keeps this computationally efficient structure and can mitigate the visual-inertial unobservable motions.

III. Range-Visual-Inertial Initialization

Given VIO unobservability issues discussed in the previous section, we present a new algorithm extending the visual-inertial initialization with a range sensor. In previous work [25], we already showed the improvements range measurements can bring to a visual-inertial filter framework. With our current approach, we extend the VINS-Mono with the ranged facet constraints. Therefore, we keep the general structure of VINS-Mono' initialization algorithm and extend it with the additional range sensor, which accounts for the new scene distance information, to a five-step algorithm as shown in Fig. 3.

A. Keyframe Selection and Initial Structure

The keyframes are selected based on a baseline criterion of [26]. If the baseline after accounting for rotation between the current image and the last keyframe exceeds a threshold th_b, the current frame is selected as the next keyframe. Further, feature tracking takes place on a frame-to-frame basis with consistent tracks developed as new frames appear.

Initially, a structure from motion (SfM) is created using the newest and oldest keyframes that exceed a baseline threshold th_b. This threshold is needed to account for hover-like motions. Then using the pose recovery criterion provided by

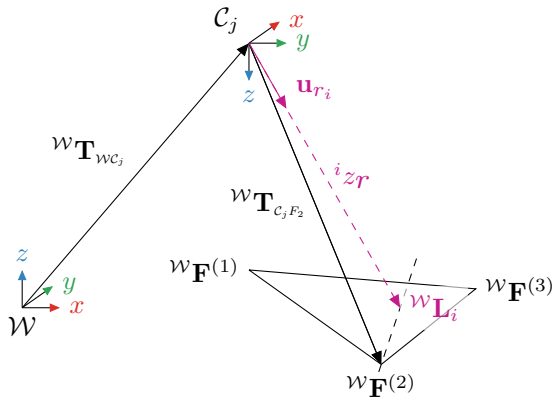

Fig. 4. The plane spanned by a Delaunay triangle which the LRF measurement intersects $\{{}^{\mathcal{W}}\mathbf{F}^{(1)}, {}^{\mathcal{W}}\mathbf{F}^{(2)}, {}^{\mathcal{W}}\mathbf{F}^{(3)}\}$ is used to to derive the estimated range ${}^{i}\hat{z}_{\boldsymbol{r}}$ from the SfM. This estimate is then compared to the LRF distance measurement ${}^{i}\breve{z}_{\boldsymbol{r}}$ to derive the metric scale for the structure and camera poses.

OrbSlam [11], the initial transformation is derived using the Fundamental or Homography matrix in the 5-point or DLT algorithm, respectively. This provides more flexibility for initialization scenarios, as it accounts for planar or structured environments. This differentiation is especially needed for downward-looking cameras, since their field-of-view more likely covers only the ground plane when flying at a low altitude.

B. Full Structure and Camera Poses

The other $N-2$ camera poses are derived using a PnP approach. First, all transforms from the initial camera pose $\mathcal{C}_k$ to all other camera poses $\mathcal{C}_j, 0 < j < N-1, j \neq k$ are derived in a *forward*-PnP. Further, any missing feature matches are triangulated. To decrease the transform calculation error between camera frames with a large baseline, a similar *backward*-PnP is performed. As a next step the newest camera pose $\mathcal{C}_k$ and all other camera poses $\mathcal{C}_j, N-1 > j > 0, j \neq k$ are used. Again all previously untriangulated feature matches between two image frames are triangulated. This vice-versa PnP is chosen for two reasons: First, this increases the number of triangulated features in the structure, which increases the amount of information available in the later bundle-adjustment stage. Second, the image overlap between the initial keyframe k and any other keyframe cannot be guaranteed. This approach tries to mitigate this issue by using the newest frame N for the transform calculation.

C. Structure Scaling

Camera only triangulation suffers from scale ambiguity. Therefore, an additional sensor is needed to scale the resulting structure of the previous step metrically. In most scenarios, the onboard IMU provides sufficient information to do so. However, in the given reference scenarios, an IMU will not yield enough metric scale information. Therefore, an additional sensor, the laser-range finder (LRF), is added to the system to provide single distance measurements at the camera rate. This range is then used to scale the structure initially.

This scaling approach follows the one proposed by Ref. [26], which models the surface structure and the range estimate as a function of the current states and measurement. However, at this point in the initialization, no state estimates are available. Therefore, only the raw, scalar distance measurements ${}^{i}\breve{z}_{\boldsymbol{r}}$ are used.

$$\begin{aligned} {}^{i}\hat{z}_{\boldsymbol{r}} &= {}^{i}\hat{z}_{\boldsymbol{r}} \cdot \frac{\mathbf{u}_{r_i}^{\mathsf{T}} \cdot \mathbf{n}}{\mathbf{u}_{r_i}^{\mathsf{T}} \cdot \mathbf{n}} \\ &= \frac{\left({}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_2} - {}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}_i}\right)^{\mathsf{T}} \cdot \mathbf{n}}{\mathbf{u}_{r_i}^{\mathsf{T}} \cdot \mathbf{n}} \end{aligned} \tag{4}$$

with

$$\mathbf{n} = \left({}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_1} - {}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_2}\right) \times \left({}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_3} - {}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_2}\right) \tag{5}$$

All tracked features from the initial triangulation frames are grouped in triangles using the Delaunay triangulation [27]. The triple of features in which the range measurement falls is selected, and its range is derived in camera frame using Equ. (4), with a visual representation shown in Fig. 4. This approach assumes a local flatness of the plane spanned by the selected triangle, an assumption that holds given enough tracked features.

Then in Equ. (6) the derived plane depth is compared to the range measurement to derive the metric scale s. This scale is then used to scale the camera poses and resulting structure metrically.

$$s = \frac{{}^{i}\hat{z}_{\boldsymbol{r}}}{{}^{i}\breve{z}_{\boldsymbol{r}}} \tag{6}$$

Please note that this derived scale is subject to the range sensor's measurement noise, feature tracker implementation, and violation of the triangle plane real-world flatness. Hence the derived scale might be error-prone. Consequently, the next step performs a range-visual optimization to minimize this initial scale error.

Further, one could argue that this scaling step can be performed before the PnP. However we chose to do this after the PnP for two reasons: First, the initial structure (A) is error prone and is minorly optimized through the PnP (B). Secondly, simulation analysis showed that scaling the structure before the R-BA (D) yields best initialization results overall.

D. Range-Visual Bundle-Adjustment

All sensors used in the above steps are subject to measurement noise. Therefore, we perform a range-visual bundle-adjustment (R-BA) optimization to reduce noise-induced measurement errors. The R-BA extends the standard bundle-adjustment with an additional term in the cost function for the LRF measurement. This addition is necessary, as the initial range measurement used for the structure scaling might be noisy and thus slightly wrong. However, adding the additional cost to the optimization reduces the impact of the assumed Gaussian white noise on the range measurement.

${}^{i}\mathbf{P}$ is the i-th image projection matrix used to project the j-th 3D-feature $\mathbf{F}^{(j)}$ onto the image plane. It is selected based on the criterion discussed in Sec. III-A. ${}^{i}\mathbf{f}^{(j)}$ is the

corresponding normalized pixel measurement in the i-th image. With this, the cost function to be minimized becomes

$$\arg\min_{{}^{i}\mathbf{P},\mathbf{F}^{(j)}} \sum_{i=0}^{N} \left(\left| {}^{i}\breve{z}_{\boldsymbol{r}} - {}^{i}\hat{z}_{\boldsymbol{r}} \right| + \sum_{j=0}^{M} d\left({}^{i}\mathbf{P}\mathbf{F}^{(j)}, {}^{i}\mathbf{f}^{(j)} \right) \right) . \quad (7)$$

E. Bias Estimation and Inertial Alignment

The IMU bias estimation from VINS-Mono is kept, which estimates the gyroscope bias using the IMU pre-integration first presented in Ref. [18]. Further, the initial acceleration bias ${}^{\mathcal{W}}\hat{\mathbf{b}}_{a} = \mathbf{0}_{\mathrm{d}}\,\mathrm{m\,s^{-2}}$ is used. Several state-of-the-art visual-inertial estimators have shown that they can handle an initial zero acceleration bias estimate and converge to the ground truth [5], [13].

The remaining initial states including only the camera frame velocities and the gravity direction, are estimated in a LLS estimation using the metrically scaled camera poses from the previous step. The frame to frame measurement matrix and vector for these remaining states are

$${}^{k}\mathbf{z}_{k+1} = \begin{bmatrix} {}^{k}\hat{\boldsymbol{\alpha}}_{k+1} - {}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} + {}^{k}\mathbf{R}_{k+1}\,{}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} - {}^{k}\mathbf{R}_{\mathcal{C}}\,\Delta^{\mathcal{C}}\mathbf{p}_{k} \\ {}^{k}\boldsymbol{\beta}_{k+1} \end{bmatrix} \quad (8)$$

$${}^{k}\mathbf{H}_{k+1} = \begin{bmatrix} -\mathbf{I}_3\,\delta t_k & \mathbf{0}_3 & \frac{1}{2}\,{}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k^2 \\ -\mathbf{I}_3 & {}^{k}\mathbf{R}_{k+1} & {}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k \end{bmatrix} \quad (9)$$

In contrast to the VINS-Mono formulation (see Eqs. (2)-(3)) the new full measurement matrix ${}^{k}\mathbf{H}_{k+N} \in \mathbb{R}^{4N\times(6N+3)}$ matrix only needs three camera poses to become invertible and the states therefore observable. Further, regardless of the scenario, the measurement vector is guaranteed to be non-zero, eliminating the possibility of the trivial solution in constant-velocity or free-fall scenarios.

IV. Simulation Tests

Initially, we investigate the performance of the proposed algorithm under the influence of standard measurement noise. Therefore, we generated range, feature, and inertial data in a point-based simulation under a constant velocity motion with ${}^{\mathcal{W}}\mathbf{v}_0 = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^{\mathsf{T}}\mathrm{m\,s^{-1}}$. All sensor noises are assumed to be white Gaussian, and are set to values representative of the sensors listed in Sec. V. We then evaluated the initialization algorithm on 100 independent Monte-Carlo runs.

The results of this Monte-Carlo simulation are displayed in Fig. 5. This figure shows the mean and standard deviation of the norm of the error in attitude, position, and velocity throughout the window. As can be seen, for all three states, the error norm is low. Especially the small estimated velocity error shows that this approach can be used to initialize a visual-inertial estimator near the optimal solution.

Furthermore, we performed various sensitivity tests with simulated data on different parameters such as (i) feature tracking pixel noise, (ii) distance measurement noise, (iii) number of keyframes in the initialization window, (iv) number of tracked features and required baseline for keyframe selection, and (v) planar and structured environments. From these tests we concluded that our algorithm performs as expected independently of the environment, with 10 keyframes in the initialization window, and with $100-200$ tracked features. The authors refer to [21] for a more detailed simulation analysis and to [28] for a stress test of the facet assumption. Further, this evaluations showed that the optimization can mitigate measurement noise if its standard deviation is below $3\,\mathrm{px}$ for the features and $10\,\%$ of the flight height for the range measurement.

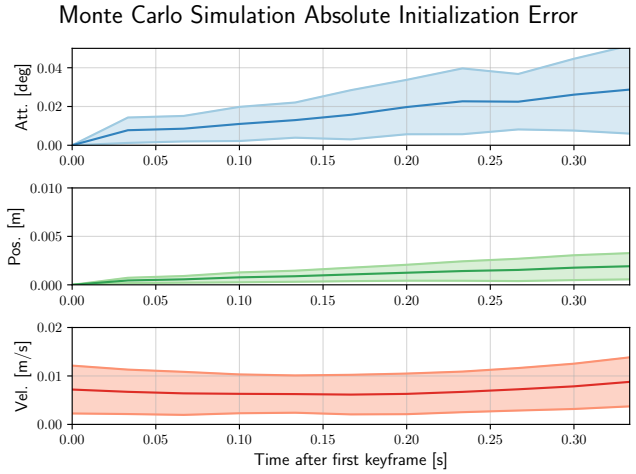

Fig. 5. Monte-Carlo evaluation of the proposed algorithm with 100 independently simulated data runs with constant velocity flight of ${}^{\mathcal{W}}\mathbf{v} = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^{\mathsf{T}}\mathrm{m/s}$ and a flight height of $1\,\mathrm{m}$. This result shows the mean and standard deviation boundaries (1σ boundary) of the error for each keyframe in the initialization window. For all runs the window was set to 10 keyframes at an image rate of $30\,\mathrm{Hz}$. The position and velocity errors throughout the initialization period is low enough to initialize a visual-inertial estimator.

V. Experiments

A. Experimental Setup

The experiments were carried out on an AscTec Hummingbird quadrocopter. Sensors included the internal IMU of the Hummingbird, a Matrixvision Bluefox mvBlueFox-MLCw camera with $640\,\mathrm{px}\times 480\,\mathrm{px}$ resolution, and a Garmin Lidar Range v3. Ground truth for all flights was recorded with an Optitrack motion capture system. The algorithm was implemented in C++, as an extension of the open-source version of VINS-Mono using the Ceres Solver [29] for the R-BA. It ran on OdroidXU4 under Ubuntu 18.04 and ROS melodic in SkiffOS [30].

In our test, the MAV was commanded to a constant velocity flight of $0.5\,\mathrm{m\,s^{-1}}$ using the Optitrack pose as reference input for the flight controller. Although inertial attitude control would be more representative of an actual mid-air re-initialization scenario, attitude and velocity control with motion capture was deemed safer to avoid a crash in the limited lab space. The constant velocity is representative of a MAV applying constant thrust and controlled to a level attitude through an IMU after a VIO failure. The initialization algorithm was started on board in real time using a window of 10 image frames with corresponding LRF measurements. The initial state estimate was then used to start the VIO navigation framework VINS-Mono. Once initialized, the reference input of the controller was switched from motion capture to VINS-Mono to demonstrate mid-air recovery and stable follow-up flight. Further, the experiments were carried out in an cluttered environment with small

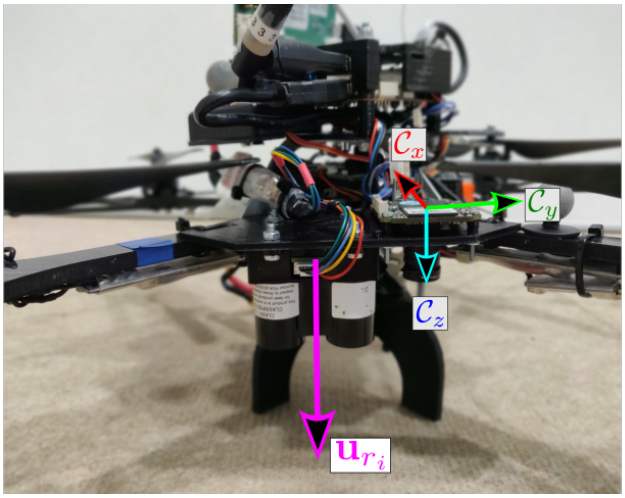

Fig. 6. For the experiments a AscTec Hummingbird quadrocopter equipped with an OdroidXU4 for onboard computations was used. The visual data (images) were provided by and Matrixvision Bluefox mvBlueFox-MLCw camera (coordinate system) mounted next to a Garmin Lidar Range v3 (pink range arrow) for single range measurements.

objects lying on a plane with a maximum height difference of $10\,\%$ of the flight height.

B. Results

The trajectory ground truth of this experiment is presented in Fig. 2. It demonstrates that our framework can initialize in a constant velocity flight condition, which would be unobservable for any VIO approach. Further, our approach can also initialize the full state of the optimization-based estimator VINS-Mono at metric scale, and then safely use it for the MAV control input. This figure further shows that our framework is accurate enough to initialize an estimator and fast enough to run onboard an embedded MAV system. For this experiment, the computation time was measured to be approximately $0.75\,\mathrm{s}$, including a data acquisition time of $0.33\,\mathrm{s}$ on the OdroidXU4 embedded computer.

Furthermore, as shown in Fig. 7, the position, velocity, and attitude error norms throughout the initialization period are low enough to initialize visual-inertial estimators. The mean and standard deviation of the error in the initialization window within this experiment is calculated to be $2.524 \pm 0.799°$ in attitude, $0.0070 \pm 0.0051\,\mathrm{m}$ in position, and $0.0794 \pm 0.0038\,\mathrm{m\,s^{-1}}$ in velocity.

We then tried to start VINS-Mono with its original initialization approach offline. Out-of-the-box VINS-Mono does not initialize in the given scenario since insufficient accelerations are present for VIO. For comparison purposes, we disabled all excitation checks in VINS-Mono and tried to initialize it under the constant velocity motion. The outcome of this test is shown in Fig. 7 (dashed lines).

In comparison to our approach, the visual-inertial initialization algorithm of VINS-Mono results in larger initial errors. Especially the unobservable metric scale in VINS-Mono's problem formulation renders it degenerate, as expected and analyzed in Sec. II-B. Subsequently, the visual-initially derived initial state led VINS-Mono to diverge as shown in Fig. 2.

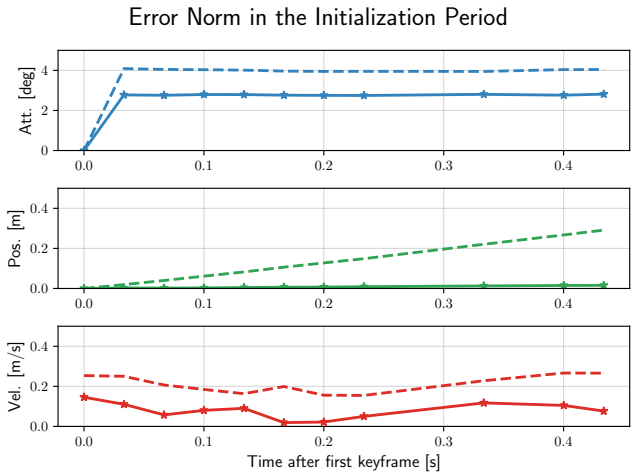

Fig. 7. Our approach's attitude, position, and velocity error norms (solid lines) of the initialization period in a real-world experiment with constant velocity flight shown in Fig. 2. In comparison, VINS-Mono's initialization state error norms are presented (dashed lines). As can be seen, our framework outperforms the visual-inertial only initialization for all three states.

VI. Conclusion

Visual-inertial odometry cannot observe the metric scale in the absence of acceleration change. This VIO limitation is even more problematic in the event of mid-air re-initialization, where either constant velocity (zero acceleration) or free-fall trajectories (constant acceleration) are expected, and other navigation states are completely unknown (unlike e.g., before take-off on the ground). We tackled this issue through a novel range-visual-inertial MAV initialization algorithm that can function even in the absence of excitation, and without prior environment nor state knowledge. As a core element of our approach, we leverage the distance measurement of a laser range finder which is tightly integrated into the visual-inertial system for robust metric system initialization in arbitrary situations. With the only requirement of local flatness (i.e., planar terrain in between three visual features) our approach is applicable in a large variety of, even to some extent cluttered, environments.

We analyzed our proposed approach in a Monte-Carlos simulation environment, which showed it to be robust against standard sensor noise values. We demonstrated our approach in real-time with closed-loop control onboard an MAV and compared it to the start-of-the-art VINS-Mono initialization algorithm. Future work includes outlier identification and rejection of the facet triangulation and full integration in an in-flight fault-detection and recovery framework.

Acknowledgment

Part of this work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 871260. Part of the research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NMOOI8D0004).

REFERENCES

[1] H. Strasdat, J. M. Montiel, and A. J. Davison, "Real-time Monocular SLAM: Why Filter?" *Proceedings - IEEE International Conference on Robotics and Automation 2010 (ICRA10)*, pp. 2657–2664, 2010.

[2] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *In Practice*, vol. 8, 2006.

[3] D. Donavanik, A. Hardt-Stremayr, G. Gremillion, S. Weiss, and W. Nothwang, "Multi-sensor fusion techniques for state estimation of micro air vehicles," in *Micro- and Nanotechnology Sensors, Systems, and Applications VIII*. Baltimore, MA: SPIE, 2016.

[4] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *IEEE International Conference on Intelligent Robots and Systems*, 2013.

[5] 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 *Proceedings - IEEE International Conference on Robotics and Automation*, 2012.

[6] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings - 2007 IEEE International Conference on Robotics and Automation (ICRA)*. Rome, Italy: IEEE, 2007, pp. 3565–3572.

[7] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[8] J. Delaune, R. Hewitt, L. Lytle, C. Sorice, R. Thakker, and L. Matthies, "Thermal-Inertial Odometry for Autonomous Flight Throughout the Night," *IEEE International Conference on Intelligent Robots and Systems (IROS) 2019*, pp. 1122–1128, 2019.

[9] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.

[10] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[11] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 10 2015.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO : Fast Semi-Direct Monocular Visual Odometry," in *IEEE International Conference on Robotics and Automation*, 2014.

[13] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1–17, 2018.

[14] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[15] S. Weiss, R. Brockers, S. Albrektsen, and L. Matthies, "Inertial Optical Flow for Throw-And-Go Micro Air Vehicles," in *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 2015, pp. 262–269.

[16] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor," in *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA: IEEE, 2015, pp. 1722–1729.

[17] M. Scheiber, J. Delaune, R. Brockers, and S. Weiss, "Visual-Inertial On-Board Throw-and-Go Initialization for Micro Air Vehicles," in *Proceedings - IEEE International Conference on Intelligent Robots and Systems (IROS) 2019*. Macau, China: IEEE, 2019, pp. 6899–6905.

[18] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation," in *Robotics: Science and Systems*, 2015.

[19] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," 0.

[20] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *IEEE International Conference on Intelligent Robots and Systems*, Vancouver, BA, 2017.

[21] M. Scheiber, "Range-Visual-Inertial Initialization For Micro Aerial Vehicles," Master's thesis, Universität Klagenfurt, Klagenfurt, Austria, 8 2020.

[22] F. R. Gantmacher and K. A. Hirsch, *The Theory of Matrices*. New York, NY, USA: Chealsea Publishing Company, 1959, vol. 1.

[23] A. Martinelli, "Closed-Form Solution of Visual-Inertial Structure from Motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[24] K. J. Wu and S. I. Roumeliotis, "Unobservable Directions of VINS Under Special Motions," Multiple Autonomous Robotic Systems Laboratory (Mars Lab), Minneapolis, MN, USA, Tech. Rep. 2, 2016.

[25] J. Delaune, R. Brockers, D. S. Bayard, H. Dor, R. Hewitt, J. Sawoniewicz, G. Kubiak, T. Tzanetos, L. Matthies, and J. B. Balaram, "Extended Navigation Capabilities for a Future Mars Science Helicopter Concept," in *2020 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, 2020, pp. 1–10.

[26] J. Delaune, D. S. Bayard, and R. Brockers, "xVIO: A Range-Visual-Inertial Odometry Framework," 2020.

[27] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Computer & Information Sciences*, vol. 9, no. 3, p. 219–242, 1980.

[28] J. Delaune, D. Bayard, and R. Brockers, "Range-visual-inertial odometry: Scale observability without excitation," *IEEE Robotics and Automation Letters*, 2021.

[29] S. Agarwal, K. Mierle, and Others, "Ceres Solver."

[30] C. Stewart, "SkiffOS: Minimal Cross-compiled Linux for Embedded Containers," Mar. 2021.