State Initialization in Visual Inertial aided Navigation robust to noisy sensor data

Jacques Kaiser[1], Agostino Martinelli[1], Flavio Fontana[2] and Davide Scaramuzza[2]

Abstract— **State of the art approaches for visual-inertial sensor fusion are filter based algorithms. These methods are recursive by design and therefore require an initialization. Due to the nonlinearity of these systems, a poor initialization can have a dramatic impact on the performance of the estimations. Recently a closed-form solution providing such an initialization has been derived. Despite mathematically sound, it is not robust to noisy sensor data in practice. In this paper, we study the impact of noisy sensor on the performance of the method. Specifically, the gyroscope bias is shown to be a major performance bottleneck. We then propose a method to automatically estimate this bias. Compared to the original method, the new method is now robust to this bias, and also provides the gyroscope bias.**

I. INTRODUCTION

Autonomous mobile robots navigating in unknown environments have an intrinsic need to perform localization and mapping using only on-board sensors. Concerning Micro Aerial Vehicles (MAV), a critical issue is to limit the number of on-board sensors to reduce weight and power consumption. Therefore, a common setup is to combine a monocular camera with an inertial measurements unit (IMU). On top of being cheap, these sensors have very interesting complementarities. Additionally, they can operate in indoor environments where Global Positioning System (GPS) signals are shadowed. An open question is how to optimally fuse the information provided by these sensors.

Currently, most sensor fusion algorithms are either filter based or iterative. That is, given a current state and measurements, they return an updated state. While working well in practice, these algorithms need to be provided by an external initial state.

The initialization of a filter based method is critical. Due to nonlinearities of the system, a poor initialization can result into converging towards local minima and providing faulty states with high confidence.

In this paper we demonstrate the efficiency of a recent closed-form solution introduced in [13][14] that fuses visual and inertial data to obtain the structure of the environment at the global scale along with the attitude and the speed of the robot. By nature, a closed-form solution is deterministic and thus does not require any initialization.

We implemented this method in order to test it with real terrain data. This allowed us to identify its bottlenecks and bring modifications to overcome them. Specifically, we investigated the impact of biased inertial measurements. Despite the case of biased accelerometer was originally studied in [14] we show that its low impact on the system makes it hard to estimate.

One major bottleneck of this method was the impact of biased gyroscope measurements. In other words, the performance becomes very poor in presence of a bias on the gyroscope and, in practice, the overall method could only be successfully used with a very precise - and expensive - gyroscope. We then introduced a simple method that automatically estimates this bias.

By adding this new method for the bias estimation to the original method we obtain results which are equivalent to the ones in absence of bias. Compared to the original method, the new method is now robust to the gyroscope bias, and also provides the gyroscope bias.

II. RELATED WORK

The problem of fusing vision and inertial data has been extensively investigated in the past. However, most of the proposed methods require an external state initialization. Because of the system nonlinearities, lack of precise initialization can irreparably damage the entire estimation process. In literature, this initialization is often guessed or assumed to be known [1][11][9][2][5].

We are therefore interested into a deterministic solution that analytically expresses the state in terms of the measurements provided by the sensors during a short time-interval.

Some deterministic solutions have been introduced in the field of computer vision and rely only on visual measurements. These techniques can recover the relative rotation and translation up to scale between two camera poses [12][8][15][7][10]. These techniques are currently used in state-of-the-art visual navigation methods on MAV in order to initialize maps [16][5]. However, the knowledge of the absolute scale and at least the absolute roll and pitch angles are essential for the MAV. It is required to take the inertial measurements into consideration to compute these values deterministically.

A closed-form solution is provided in [3] to determine the absolute scale. However, it needs an additional onboard sensor measuring a metric quantity, such as an altimeter or an air pressure sensor. It is therefore not applicable for our minimal configuration.

A procedure to quickly re-initialize a MAV after a failure is discussed in [4]. However, their method also requires an altimeter to estimate the absolute scale.

A landmark of known dimension is used in [6] to recover the initial pose of the MAV. This method is therefore not suited to unknown environment.

Recently, a closed-form solution has been introduced in [13]. From integrating inertial and visual measurements over a short time-interval, this solution provides the absolute scale, roll and pitch angles, initial velocity and distance to features. Specifically, all the physical quantities are obtained by simply inverting a linear system. The solution of the linear system can be refined with a quadratic equation assuming the knowledge of the gravity magnitude.

This closed-form has been improved in [11] to work with unknown camera-IMU calibration. Specifically, the translation between the camera and the IMU is expressed as an unknown in the linear system. However, the relative rotation between the camera and the IMU can not be expressed this way, thus an alternative way to compute it is proposed. This method is therefore independent of external camera-IMU calibration, hence well suited for power-on-and-go systems.

A more intuitive expression of the same closed-form solution is derived in [14]. While being mathematically sound, this closed-form solution is not robust to noisy sensor data. For this reason, to the best of our knowledge, it has never been used in an actual application. In this paper we carry out an analysis to find out its limitations. Specifically, we show that it is resilient to the accelerometer bias but strongly affected by the gyroscope bias. We then introduce a simple method that automatically estimates the gyroscope bias. By adding this new method for the bias estimation to the original method we obtain results which are equivalent to the ones in absence of bias. Compared to the original method, the new method is now robust to the gyroscope bias, and also provides the gyroscope bias. We validate our new method against real terrain data to prove its robustness against noisy sensors.

III. THE CLOSED-FORM SOLUTION

In this paper, we do not provide a new derivation of the closed-form solution. Instead, we consider the latest derivation proposed in [14]. Specifically, let us refer to a short interval of time (of the order of 3 seconds). We assume that during this interval of time the camera observes simultaneously N point-features and we denote by $t_1,\ t_2,\cdots,t_{n_i}$ the times of this interval when the camera provides an image of these points. Without loss of generality, we can assume that $t_1 = 0$. The following equation holds (see [14] for its derivation):

$$S_j = \lambda_1^i \mu_1^i - V t_j - G\frac{t_j^2}{2} - \lambda_j^i \mu_j^i \qquad (6)$$

With:

- μ_j^i the normalized bearing of point feature i at time t_j in the local frame at time t_1;
- λ_j^i the distance to the point feature i at time t_j;
- V the velocity in the local frame at time t_1;
- G the gravity in the local frame at time t_1;
- S_j the integration in the interval $[t_1,\ t_j]$ of the rotated linear acceleration data (i.e., the integration of the inertial measurements).

The local frame refers to a frame of reference common to the IMU and the camera. In a real application, we would work in the IMU frame and have some additional constant terms accounting for the camera-IMU transformation. We do not express these constant calibration terms explicitly here for clarity reasons.

The unknowns of Equation 6 are the scalars λ_j^i and the vectors V and G. Note that the knowledge of G is equivalent to the knowledge of the roll and pitch angles. The vectors μ_j^i are fully determined by visual and gyroscope measurements [1], and the vectors S_j are determined by accelerometer and gyroscope measurements.

Equation 6 provides three scalar equations for each point feature $i = 1,...,N$ and each frame starting from the second one $j = 2,...n_i$. We therefore have a linear system consisting of $3(n_i - 1)N$ equations in $6 + Nn_i$ unknowns. Indeed, note that when the first frame occurs, at $t_1 = 0$, Equation 6 is always satisfied thus does not provide information. We can write our system using matrix formulation. Solving the system is equivalent to inverting a matrix of $3(n_i - 1)N$ rows and $6 + Nn_i$ columns.

In [14], the author proceeded to one more step before expressing the underlying linear system. For a given frame j, the equation of the first point feature $i = 1$ is

[1] The gyroscope measurements in the interval $[t_1,\ t_j]$ are needed to express the bearing at time t_j in the frame at time t_1

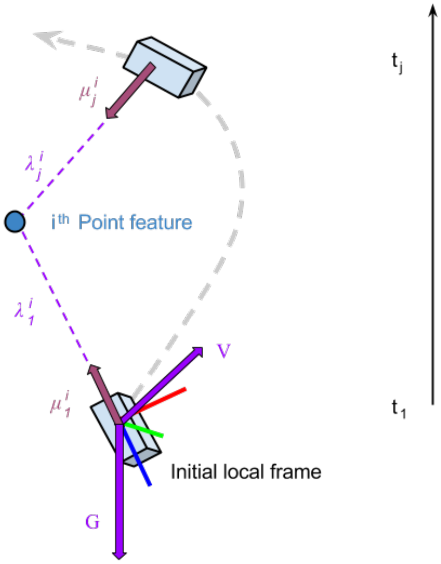


Fig. 1: Visual representation of Equation 6. The unknowns of the equation are colored in purple.

subtracted to all other point features equation $1 < i <= N$ concerning frame j (Equation 7). This additional step, very useful to detect system singularities, has the effect to corrupt all measurements with the first measurement, hence worsening the performance of the closed-form solution. In this paper, we do not take to this additional step.

The linear system in Equation 6 can be written in the following compact format:

$$\Xi X = S \tag{9}$$

The matrix Ξ and the vector S are fully determined by the measurements, while X is the unknown vector. We have:

$$S \equiv [S_2^T, ..., S_2^T, S_3^T, ..., S_3^T, ..., S_{n_i}^T, ..., S_{n_i}^T]^T$$

$$X \equiv [G^T, V^T, \lambda_1^1, ..., \lambda_1^N, ..., \lambda_{n_i}^1, ..., \lambda_{n_i}^N]^T$$

$$\Xi \equiv$$

$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
T_2 & S_2 & \mu_1^1 & 0_3 & 0_3 & -\mu_2^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
T_2 & S_2 & 0_3 & \mu_1^2 & 0_3 & 0_3 & -\mu_2^2 & 0_3 & 0_3 & 0_3 & 0_3 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
T_2 & S_2 & 0_3 & 0_3 & \mu_1^N & 0_3 & 0_3 & -\mu_2^N & 0_3 & 0_3 & 0_3 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
T_{n_i} & S_{n_i} & \mu_1^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^1 & 0_3 & 0_3 \\
T_{n_i} & S_{n_i} & 0_3 & \mu_1^2 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^2 & 0_3 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
T_{n_i} & S_{n_i} & 0_3 & 0_3 & \mu_1^N & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^N
\end{array}\right]$$

Where $T_j \equiv -\frac{t_j^2}{2} I_3$, $S_j \equiv -t_j I_3$ and I_3 is the identity 3 x 3 matrix; 0_{33} is the 3 x 3 zero matrix. Note that the matrix Ξ and the vector S are slightly different from the one proposed in [14]. This is due to the additional step we did not take for numerical stability reasons.

The sensor information is completely contained in the above linear system. Additionally, in [14], the author added a quadratic equation assuming the gravitational acceleration is a priori known. Let us denote the gravitational magnitude by g. We have the extra constraint $|G| = g$. We can express this constraint in matrix formulation:

$$|\Pi X|^2 = g^2 \tag{10}$$

With $\Pi \equiv [I_3, 0_3, ..., 0_3]$. We can therefore recover the initial velocity, the roll and pitch angles and the distances to the point features by finding the vector X which satisfies 9 and 10.

In the next sections, we will evaluate the performance of this method on real terrain data. This will allow us to identify its weaknesses and bring modifications to overcome them.

IV. PERFORMANCE BOTTLENECKS

A. Test setup

The MAV performs a motion while being tracked with an optical Vicon system. We can therefore compare our estimations with the ground truth. We define the relative error as the euclidean distance between the estimation and the ground truth, normalized by the ground truth. We measure our error on the absolute scale by computing the mean error over all estimated distances to point features λ_j^i.

To identify the performance bottlenecks, we used IMU data obtained from terrain acquisitions while we simulated the visual measurements. This separation allowed us to know the ground truth for the distance to the point features and also better understand the weaknesses

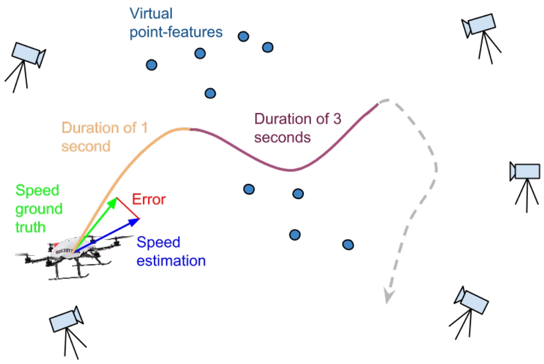


Fig. 2: Test setup for identifying the performance bottlenecks. The drone is equipped with an IMU, and the visual measurements are simulated. It performs a motion while being tracked by a Vicon system.

of our method. Note that in the final experiment (see section VI) we use real visual measurements.

We represent this setup in Fig. 2.

In general, we use one frame every 0.1 seconds, even if the camera provides significantly more frames. Indeed, we can discard most of the frames provided by the camera. If two frames are too close to each other, then the additional equations do not bring much information to our system. Reducing the number of considered frames reduces the size of the matrices, thus speeds up the computations.

As an example, over a time interval of 3 seconds, we obtain 31 distinct frames. When observing 7 features, it yields a system of $3\times 30\times 7 = 630$ equations and $6+7\times 31 = 223$ unknowns.

The method we use to solve the overconstrained linear system $\Xi X = S$ is a singular value decomposition (SVD) since it yields numerically robust solutions.

In this section, we will start by presenting the results obtained with the original closed-form solution on terrain IMU data. Our goal is to identify its performance bottlenecks and introduce modifications to overcome them.

B. Original closed-form solution performance

The original closed-form solution described in Equation 9 will be used as a basis for our work. Moreover, we can also use the knowledge of the gravity magnitude in order to refine our results with Equation 10. In this case, we are minimizing a linear objective function with a quadratic constraint. in Fig. 4, we represent the quality of the evaluations with and without this additional constraint.

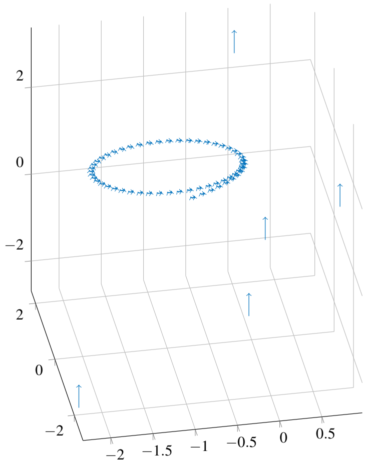


Fig. 3: Motion performed by the drone in 5 seconds.

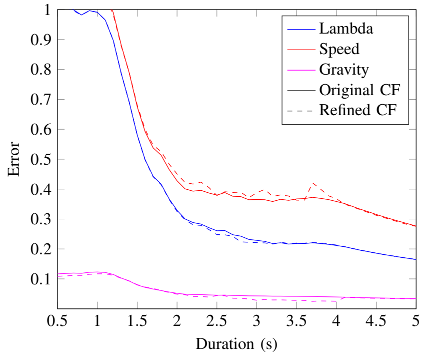
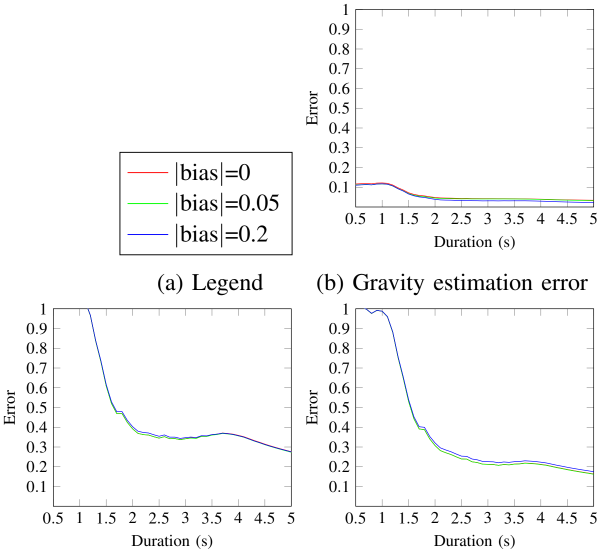


Fig. 4: Original closed-form solution estimations with and without gravity knowledge refinement. We are observing 12 features over a variable duration of integration.

Note how the evaluations get better as we increase the integration duration. Indeed, our equations come from an extended triangulation [13]. Therefore, it requires a significant difference in the measurements over time to robustly estimate the state. Also note that the gravity is robustly estimated (around 5% error), whereas the speed and the distance to the features is more erroneous (above 10% error).

Since the gravity is well estimated with the original closed-form, it comes without surprise that constraining its magnitude does not improve the performance much. That is why the curves representing the performance with and without the gravity refinement are so close.

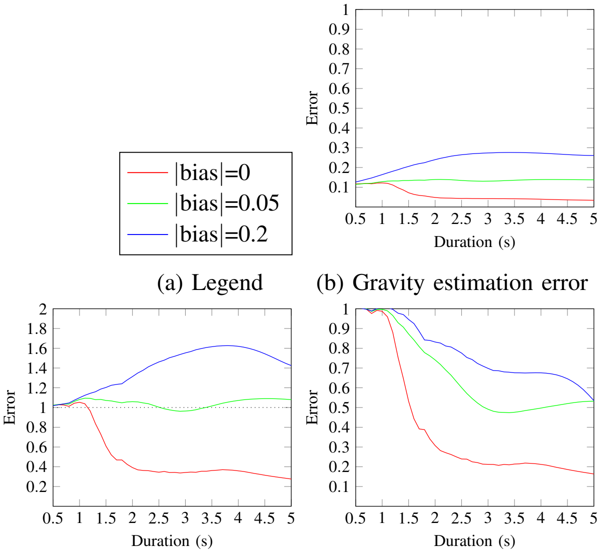
In the following sections, we will study the impact of biased inertial measurements on the performance of the closed-form solution without considering the gravity



(a) Legend (b) Gravity estimation error

(c) Velocity estimation error (d) Lambda estimation error

Fig. 5: Impact of the accelerometer bias on the performance of the closed-form solution. We are observing 7 features over a variable duration of integration.



(a) Legend (b) Gravity estimation error

(c) Velocity estimation error (d) Lambda estimation error

Fig. 6: Impact of the gyroscope bias on the performance of the closed-form solution. We are observing 7 features over a variable duration of integration.

refinement.

C. Impact of accelerometer bias on the performance

In order to visualize the impact of the accelerometer bias on the performance, we corrupt the accelerometer measurements provided by our terrain IMU by adding an artificial bias (Fig. 5).

Despite a high accelerometer bias the closed-form solution still provides robust results.

As seen on the Fig. 5, neither the estimation of the gravity, the velocity or the lambdas are impacted by the accelerometer bias.

In [14], the author provides an alternative formulation of the closed-form solution including the accelerometer bias as an observable unknown of the system. However, the estimation of the accelerometer bias with this method is not robust since our system is only slightly affected by it.

This is a counterintuitive result. Since our equations contain an integration of the acceleration, we also perform an integration of the accelerometer bias. We would have expected the accelerometer bias to have a greater impact on the solutions yielded by the system.

D. Impact of gyroscope bias on the performance

Again, in order to visualize the impact of the gyroscope bias on the performance, we corrupt the gyroscope measurements provided by our terrain IMU by adding an artificial bias (Fig. 6).

Our experiments have shown that the presence of gyroscope bias significantly damages the results of the closed-form solution.

As seen in Fig. 6, the performance becomes very poor in presence of a bias on the gyroscope and, in practice, the overall method could only be successfully used with a very precise—and expensive—gyroscope.

V. ESTIMATING THE GYROSCOPE BIAS

Previous work has shown that the gyroscope bias is an observable mode when using an IMU and a camera, which means that it can be estimated [13].

Optimally, we would add the gyroscope bias in our unknown vector X and determine X by simply inverting the system $\Xi X = S$ as in the standard closed-form solution. However, we can not express the gyroscope bias linearly with this system.

In this section, we propose a different approach to estimate the gyroscope bias using the closed-form solution.

A. Nonlinear minimization of the residual

Since our system of equations (6) is overconstrained, inverting it is equivalent to finding the vector X that minimizes the residual $||\Xi X - S||^2$.

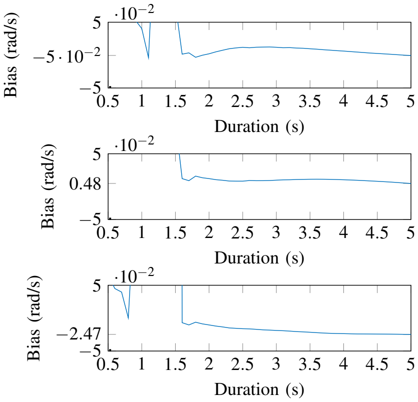


Fig. 7: Gyroscope bias estimation from nonlinear minimization of the residual. We are observing 12 features over a variable duration of integration.

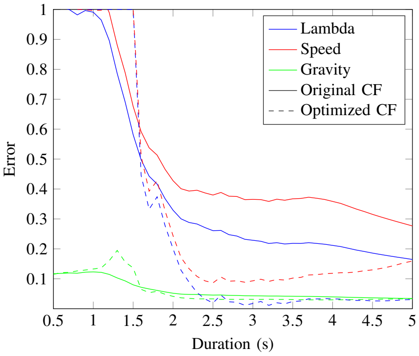


Fig. 8: Estimation error of the original closed-form solution against the optimized closed-form solution. We are observing 12 features over a variable duration of integration.

Since we cannot express the gyroscope bias linearly, we define the following cost function:

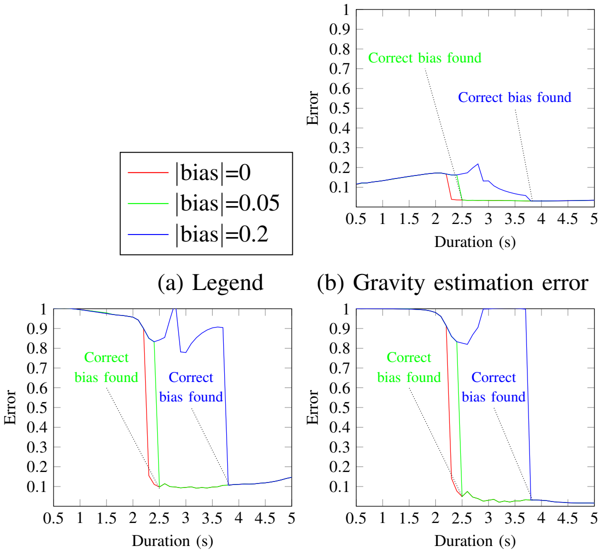
$$cost(B) = ||\Xi X - S||^2 \tag{1}$$

With:

- B the gyroscope bias;
- Ξ and S computed with respect to B.

By minimizing this cost function, we recover the gyroscope bias B and the unknown vector X which compensated for the gyroscope bias B. We can initialize the optimization process with $B = 0_3$ since the bias is usually a rather small quantity. Figure 7 displays the three components of the bias estimated by this minimization.

The optimized closed-form solution provides better results than the standard closed-form solution. Fig. 8 depicts an improvement in precision of around 13% for the distance to the features, and around 30% for the speed after 4 seconds of integration.



(a) Legend (b) Gravity estimation error

(c) Velocity estimation error (d) Lambda estimation error

Fig. 9: Impact of the gyroscope bias on the performance of the optimized closed-form solution. We are observing 7 features over a variable duration of integration.

Moreover, this method is robust even for high values of the gyroscope bias. Fig. 9 represents the quality of the estimations with the same artificial gyroscope bias from Fig. 6.

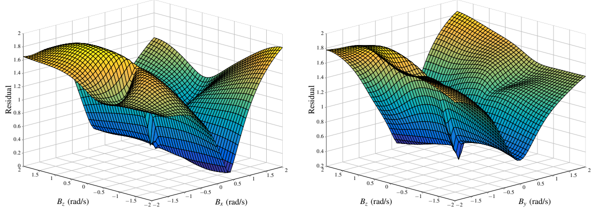
As seen in Fig. 9, after a certain integration duration, the estimations agree no matter how high the bias is. In other words, given that the integration duration is long enough, this method is unaffected by the gyroscope bias.

However, for very short time of integration (< 1.7 seconds), the gyroscope bias can be estimated to large and unlikely values. To understand this misestimation, in Fig. 10 we plot the residual with respect to the bias, which is the cost function we are minimizing. We highlight a misestimation of the gyroscope bias by setting the duration of integration to 2 seconds while observing 7 features. We refer to the components of the gyroscope bias by $B = [B_x, B_y, B_z]$.

As we can see in Fig. 10, the cost function admits a symmetry with respect to B_z.

B. Removing the symmetry in the cost function

The symmetry in the cost function is induced by the strong weight of the gravity in the equation 6. In general, the residual is almost constant with respect to the component of the gyroscope bias along the direction $\vec{u}$ when



(a) Residual with respect to B_x and B_z

(b) Residual with respect to B_y and B_z

Fig. 10: Cost function (residual) with respect to the gyroscope bias for a small amount of available measurements (integration of 2 seconds while observing 7 features)

this direction $\vec{u}$ is collinear with the gravity throughout the motion. In the terrain data we had, the motion satisfies this constraint. Specifically, the gyroscope was strapped on the MAV such that the vector $[0,0,1]$ in the gyroscope frame was pointing upwards when the MAV was hovering. That is why the residual varies only slightly for certain time sequences with respect to this vector. Indeed, in normal operations, a MAV will often have a pose close to its hovering stance in order to stay stable.

If the MAV rotates such that the vector $\vec{u}$ becomes noncollinear with the gravity, the cost function does not exhibit this symmetry anymore. In this case the gyroscope bias is well estimated.

A simple solution to avoid having that symmetry in our system would be to constrain the motion of our MAV while it is operating. Another way to artificially get rid of this symmetry is to tweak the cost function. Specifically, we can add a regularization term that penalizes high estimations of the gyroscope bias:

$$cost(B) = ||\Xi X - S||^2 + \lambda ||B|| \tag{2}$$

The coefficient λ is the weight given to how much we want the bias to be small. For small values of λ, our cost function is similar to the previous one and the bias can grow arbitrarily high. For high values of λ, the estimations provided by the optimized closed-form solution are similar to the ones provided by the standard closed-form solution. Indeed, high values of λ force the estimation of the bias to 0_3.

Note that, instead of forcing the gyroscope bias to be close to 0_3, we can easily force it to be close to any value. Therefore, if we have the knowledge of an approximately known gyroscope bias, we can use it to provide a better estimation of the gyroscope bias.

$$cost(B) = ||\Xi X - S||^2 + \lambda ||B - B^{approx}||$$

With B^{approx} the known approximate gyroscope bias. This methods allow us to reuse previously computed gyroscope bias since it is known to slowly vary over time.

Selecting a reliable and safe value for the regularization parameter λ is complicated. In this paper, we picked a value of λ by experimentation.

VI. VALIDATION

We validate our method against a different dataset than the one we used in the previous sections to draw our conclusions. Specifically, it contains IMU and camera measurements along with ground truth. Therefore, we are now only relying on fully real terrain data. However, since we are no longer simulating the point features, we do not have the ground truth for the distance to the point features anymore. We can therefore only compare the performance of the evaluation of the speed and the gravity.

The drone is flying indoor at low altitude. The feature extraction and matching is done with FAST corner algorithm as in [5].

We compare the performance on the estimations of the gravity and the initial velocity obtained with three different methods:

- The original closed-form solution (Equation 9);
- Our modified closed-form solution (Equation 1);
- Fast Semi-Direct Monocular Visual Odometry (SVO) described in [5].

The reason we included the SVO in the validation is for having a reference to state of the art pose estimation method. However, this method requires to be initialized with the knowledge of the absolute scale, whereas our method works without initialization.

We set the integration duration for the closed-form solution to 2.8 seconds. The camera provides 60fps, but we discard most of the frames and consider only one frame every 0.1 seconds. Indeed, considering frames that are too close to each other does not add significant information to our system.

As seen in Fig. 11, the original closed-form solution and the optimized closed-form solution have similar performance. Indeed, for this dataset the gyroscope bias was estimated to $B = [0.0003, 0.009, 0.001]$, which is very low ($||B|| = 0.0091$).

Moreover, the performance is also similar to the one obtained with a well-initialized SVO. We remind the reader that unlike SVO, the closed-form solution does not require the knowledge of the absolute scale to be provided.

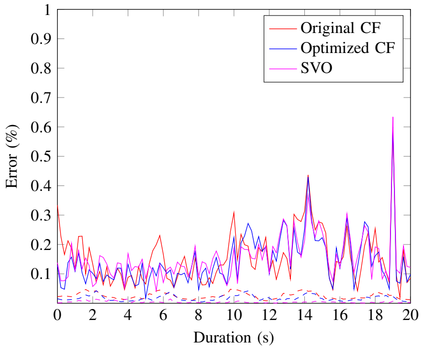


Fig. 11: Estimation error of the optimized closed-form solution against the original closed-form solution [14] and SVO [5]. The duration of integration is set to 2.8 seconds, and 10 point features are observed throughout the whole operation.

VII. Conclusion

In this paper, we have studied the recent closed-form solution proposed by [14] that performs visual-inertial sensor fusion without requiring an initialization. We implemented this method in order to test it with real terrain data. This allowed us to identify its performance bottlenecks and bring modifications to overcome them.

We investigated the impact of biased inertial measurements. Despite the case of biased accelerometer was originally studied in [14] we show that its low impact on the system makes it hard to estimate.

One major performance bottleneck of this method was the impact of biased gyroscope measurements. In other words, the performance becomes very poor in presence of a bias on the gyroscope and, in practice, the overall method could only be successfully used with a very precise - and expensive - gyroscope. We then introduced a simple method that automatically estimates this bias.

We validated this method by comparing its performance against the original method and the SVO described in [5] which is the state of the art approach for pose estimation on MAV.

For future work, we see this optimized closed-form solution being implemented on a MAV to provide accurate state initialization. This would allow aggressive take-off maneuvers, such as hand throwing the MAV in the air [4]. With our technique however, we could get rid of the altimeter sensor. The drone could therefore perform any motion right after the throw instead of having a required hovering stage to compute the absolute scale.

References

[1] L. Armesto, J. Tornero, and M. Vincze. Fast Ego-motion Estimation with Multi-rate Fusion of Inertial and Vision. *The International Journal of Robotics Research*, 26(6):577–589, 2007.

[2] Marco Bibuli, Massimo Caccia, and Lionel Lapierre. Sliding Window Filter with Application to Planetary Landing. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 7(PART 1):81–86, 2007.

[3] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1456, 2013.

[4] Matthias Faessler, Flavio Fontana, Christian Forster, and Davide Scaramuzza. Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor. In *International Conference on Robotics & Automation*, 2015.

[5] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[6] P. Gemeiner, P. Einramhof, and M. Vincze. Simultaneous Motion and Structure Estimation by Fusion of Inertial and Vision Data. *The International Journal of Robotics Research*, 26(6):591–605, 2007.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[9] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. On the complexity and consistency of UKF-based SLAM. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4401–4408, 2009.

[10] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. *Proceedings - International Conference on Pattern Recognition*, 1:630–633, 2006.

[11] M. Li and a. I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[12] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections, 1981.

[13] Agostino Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *Transactions on Robotics*, 28(1):44–60, 2012.

[14] Agostino Martinelli. Closed-form solution of visual-inertial structure from motion. *International Journal of Computer Vision*, 106(2):138–152, 2014.

[15] D Nister. An efficient solution to the five point relative pose problem. pages 195–202, 2003.

[16] Stephan M Weiss. Vision Based Navigation for Micro Helicopters (PhD Thesis - Weiss 2012). (20305), 2012.