Invariant Filter Based Preintegration for Addressing the Visual Observability Problem

Yehonathan Litman
Department of Computer Science
Stony Brook University
Stony Brook, New York
yehonathan.litman@stonybrook.edu

1Ya Wang, 2Libo Wu
1Department of Mechanical Engineering
2Department of Electrical and Computer Engineering
Texas A&M University
College Station, Texas

Abstract—tightly-coupled Visual Inertial Navigation Systems (VINS) implementations have proven their superiority due to their ability to jointly optimize all state variables. However, many approaches are based upon an Extended Kalman Filter (EKF) for optimization given their simplicity and efficiency, but for that very reason suffer from inconsistencies associated with state uncertainty. This is due to the observability of the system which is not invariant for all transformations when state is filtered through an EKF. In this paper, we address this issue through the use of an Invariant Extended Kalman Filter (I-EKF) to address the observability challenge. We derive an IMU factor that does not depend on the assumption that the biases are the same between two sequential keyframes and that the VINS, when given the preintegrated measurements, does not have to separately estimate the state variables. After integrating our algorithm with the open-source parallel tracking and mapping system ORB-SLAM [1], experimental results confirm that our derivation is computationally efficient while demonstrating superior accuracy compared to state-of-the-art filtering and graph optimization based VINS.

I. INTRODUCTION

The Simultaneous Localization and Mapping (SLAM) challenge has been approached from multiple directions with multiple methodologies, but for small platforms such as mobile phones and Micro Aerial Vehicles (MAVs), Visual Inertial Navigation Systems (VINS) have proven their superiority due to their low physical requirements. Using a synchronized monocular camera with an inexpensive IMU, VINS allow for generating accurate odometry in GPS-denied environments with a minimal, yet highly practical setup. In traditional vision-based systems, a monocular (or stereo) camera attempted to track its’ position and orientation in space through either optimization or filtering. Since these systems were fundamentally sensitive to rotations and fast movements, the addition of an IMU made it possible to keep better track of the because it allows for recovering the global roll, pitch and the undetermined scale (for monocular setups) all while dramatically improving tracking performance by bridging the gaps where visual sensing is lost.

A number of methods regarding VINS have been developed with varying effectiveness and practicability. Initial implementations were of loosely coupled approaches [2][3] where the IMU was considered an independent unit from vision based SLAM. The IMU was used for propagating the state while pose tracking was done using the camera. Quickly, tightly-coupled implementations [4][5][6] proved their superiority, as they use only one estimator to find the optimal state estimate by fusing raw measurements from the camera and IMU directly. One type of VINS is based on optimization of variables based on their history, and while tightly-coupled optimization approaches are very accurate, the computational complexity grows steadily over time, as each batch of measurements must be re-linearized when the system optimizes its state. As a result, a great number of visual inertial approaches are based on a simple yet effective Extended Kalman Filter (EKF) to process a measurement only once whenever the state updates. Despite their successes, it has been shown that EKF based VINS suffer from variations linked to the state uncertainty [7]. This is due to the constrained observability of these VINS given linear errors arising from the EKF based optimization [8]. Algorithms such as multi-state constrained Kalman filters serve to improve the consistency and accuracy of the system by correcting against the aforementioned weakness, but through the cost of extra computations as well as ignoring older states [9].

To enhance performance many VINS, including EKF based ones, have recently began turning to manifold and Lie group representations to increase the tracking accuracy [10][11]. Their use allows for easier Jacobian manipulations, all while avoiding the Euler singularity issue. Unfortunately, these methods still suffer from the lack of transformation symmetry about the observability due to the use traditional EKFs and thus fail to fully reap the benefits of manifold implementations. This is where an Invariant Extended Kalman Filter (I-EKF) based system can mitigate the observability challenge, since it can preserve symmetry and guarantee local convergence towards the local state [12][13]. The IMU factor is continuous, which allows for high-order integration as well as discerning state uncertainties between two keyframes [14]. A simple nonlinear least square approach is able to optimize and correct the system’s state against the IMU noise.

Lastly, to fully exploit a large scale operation, it is important not to marginalize the old states for the sake of accuracy and to avoid inconsistencies with state estimates [15]. We can consider graph based optimization for our tracking system such as an incremental optimization strategy (iSAM) but with full smoothing where early observations are not ignored [10].
However, the role of physical CPU limitations needs to be considered as the need for memory and computation power grow over time. We can address this by using fixed lag smoothing, but this decreases accuracy as older states are once again disregarded. The most proper solution is to use a parallel system that continuously associates the back-end module with the front-end module and can support an implementation where older states are not disregarded, the most appropriate of which is ORB-SLAM [1]. This parallel implementation is critical for our optimization needs, and ORB-SLAM’s efficient use of bundle adjustment for feature tracking makes it an excellent candidate as opposed to systems which use other types of feature tracking such as optical flow which is inaccurate at tracking when presented with low-quality image streams [11].

II. IMU Factor and Preintegration

A. State Formulation

The state of the IMU that needs to be estimated can be represented by the following orientation, position, velocity, and biases at the k-th local frame

\[
\mathbf{x} = [\mathbf{R}_k, \mathbf{p}_k, \mathbf{v}_k, \mathbf{b}_k]
\]

(1)

Where the pose \((\mathbf{R}_k, \mathbf{p}_k) \in \text{SE}(3), \mathbf{v}_k = \dot{\mathbf{p}}_k \in \mathbb{R}^3\) is the velocity of the IMU in the inertial frame, and the biases are denoted by \(\mathbf{b}_k = [\mathbf{b}_k^g, \mathbf{b}_k^a] \in \mathbb{R}^6\), where \(\mathbf{b}_k^g, \mathbf{b}_k^a \in \mathbb{R}^3\) represent the gyro and accelerometer white Gaussian noise bias, respectively. At the k-th frame, both the gyroscope and the accelerometer measurements \(\tilde{\mathbf{w}}_k\) and \(\tilde{\mathbf{a}}_k\) inside the IMU sensor are affected by the white Gaussian noise bias, in addition to a slowly varying random walk bias

\[
\tilde{\mathbf{w}}_k = \mathbf{w}_k + \mathbf{b}_k^g + \mathbf{n}_k^g
\]

(2)

\[
\tilde{\mathbf{a}}_k = \mathbf{R}_{\mathbf{w}b}^r(\mathbf{a}_k - \mathbf{g}_w) + \mathbf{b}_k^a + \mathbf{n}_k^a
\]

(3)

Where \(\mathbf{w}_k\) and \(\mathbf{a}_k\) are the true gyro and accelerometer states at the k-th frame, \(\mathbf{g}_w\) is the gravitational vector at the inertial frame coordinates, and \(\mathbf{n}_k^g\) and \(\mathbf{n}_k^a\) are the slowly varying random walk biases where \(\mathbf{n}_k^g = \mathbf{b}_k^g\) and \(\mathbf{n}_k^a = \mathbf{b}_k^a\). The prefixes B and W correspond to the IMU body coordinate frame and the inertial frame respectively. All variables refer to the transformation from the inertial frame to the body frame unless specified otherwise. The biases \(\eta = (\mathbf{b}_k^g, \mathbf{b}_k^a, \mathbf{n}_k^g, \mathbf{n}_k^a)\) all have an IMU noise covariance constant matrix \(\mathbf{Q} \in \mathbb{R}^{12}\) as prior known knowledge.

B. Motion Integration

In order to characterize the motion model of the system, we use the following kinematic model for the IMU-driven estimation motivated from the works in [17][18]

\[
\dot{\mathbf{R}}_k = \mathbf{R}_k \dot{\mathbf{w}}_k^g, \quad \dot{\mathbf{R}}_k = \mathbf{R}_k \mathbf{a}_k + \mathbf{g}_w, \quad \dot{\mathbf{p}}_{k}^{wb} = \mathbf{v}_{k}^{wb}\]

(4)

Where the notation \(^\wedge\) stands for the skew symmetric matrix operation. A simple integration operation between a few frames, where we are given a state \(\mathbf{x}_i\) at frame \(i\) and several IMU measurements \(\mathbf{a}_{ij}\) and \(\mathbf{w}_{ij}\), can naively predict the future state of our system \(\mathbf{x}_j\). However, simply integrating between frames of measurement is computationally expensive, thus we use a quantification of the IMU state error, where, since an invariant EKF is an extension of an EKF, we can represent the following [12]

\[
\mathbf{x} = \tilde{\mathbf{x}} \oplus \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})
\]

(5)

Where \(\mathbf{e}\) corresponds to the white Gaussian noise vector and the notation \(\oplus\) alludes to the retraction process [19]. To derive the inverse of retraction, we recall some useful identities associated with the Special Orthogonal Group SO(3) and the Special Euclidean Group SE(3) [10]. We first define the exponential map (at the identity) that associates an element of the Lie Algebra to a rotation

\[
\exp(\varphi) = \mathbf{I}^T (\mathbf{R} - \mathbf{R}^T) + \frac{\sin(\|\varphi\|)}{\|\varphi\|} \varphi^T (\mathbf{R}^T - \mathbf{I}) + 
\]

(6)

By associating a rotation matrix \(\mathbf{R}\) in SO(3) to a skew symmetric matrix, we calculate the logarithm map

\[
\log(\mathbf{R}) = \frac{\varphi (\mathbf{R} - \mathbf{R}^T)}{2 \sin(\varphi)} , \quad \varphi = \cos^{-1}
\]

(7)

Using this information, we can derive the inverse mapping operation \(\Theta\) of the retraction operation. The state uncertainty thus equals

\[
\mathbf{e} = \mathbf{x} \oplus \tilde{\mathbf{x}} = \left[ \begin{array}{c} \log(\mathbf{R})^T \mathbf{R} \\ \mathbf{R}^T (\tilde{\mathbf{p}} - \mathbf{p}) \\ \mathbf{b}_g - \tilde{\mathbf{b}}_g \\ \mathbf{b}_a - \tilde{\mathbf{b}}_a \end{array} \right]
\]

(8)

We can now calculate the propagation model from the linearized error-state equation

\[
\dot{\mathbf{e}} = \mathbf{F} \mathbf{e} + \mathbf{G} \eta
\]

(9)

Where \(\mathbf{F}\) and \(\mathbf{G}\) are the time varying Jacobian matrices. The state transition matrix \(\Phi_n = \Phi(k_n, k_{n+1})\) is computed as follows

\[
\frac{\partial}{\partial k} \Phi(k, k_n) = \mathbf{F}_k \Phi(k, k_n) , \quad \Phi(k_n, k_n) = \mathbf{I}
\]

(10)

Now the covariance \(\mathbf{Q}\) of \(\mathbf{e}\) can be estimated through small scale integration

\[
\dot{\mathbf{Q}}_n = \int_{k_n}^{k_{n+1}} \Phi(k_n, k_{n+1}) \mathbf{Q} \Phi^T(k_{n+1}, k_n) \, \mathbf{Q} \, \mathbf{Q}^T \, \Phi(k_n, k_{n+1}) \, \partial \tau
\]

(11)

As for the invariance of the filter, we can estimate the true state via a retraction operation shown in Eq. (5). It can then be proven that the I-EKF is indeed invariant under both stochastic and deterministic transformations, while a general EKF is invariant for deterministic transformations but not for stochastic ones (for...
a complete proof, see [12]). In the case of our VINS, deterministic unobservable transformations refer to translation in the three dimensions, while a stochastic unobservable transformation refers to yawing around the gravitational axis about the inertial frame.

C. IMU Factor
In order not to leave out any old states, we implement graph optimization for the IMU state with initial estimated covariance \( \tilde{Q} = 0 \). In correspondence with the invariance of our system, the cost function between frames \( i \) and \( j \) is

\[
\delta(x_i, x_j) = x_j \otimes \begin{bmatrix} R_{i,j}^L \\ R_{i,j}^E \end{bmatrix}
\]

Where all variables other than the rotation factors are linear. Lastly, to merge the front-end and back-end of our algorithm, the projection function of ORB-SLAM was changed to accommodate the transformation \( T_{CB} \) from the camera to the body of the IMU with a simple modification to the original projection function

\[
e_{i,j} = c_{i,j} - T_{CB} \pi_i(T_{CB}, C_j)
\]

Where \( c_{i,j} \) is a two-dimensional matched feature point \( j \) at keyframe \( i \), \( T_c \) is the keyframe pose with respect to the inertial frame, \( \pi_i \) is the projection function for keyframe \( i \), \( C_j \) are the three-dimensional feature locations, and \( e_{i,j} \) is the error for the observation of a map point \( j \) in keyframe \( i \).

III. MAXIMUM A POSTERIORI VISUAL-INERTIAL ESTIMATION
For our back-end calculations, it is assumed that the transformation from the camera to the IMU body is already known, the camera is calibrated as a pinhole model, and that the front-end is providing three-dimensional feature points found in the image, as well as keyframes. Our system is not structureless as it depends on feature measurements for estimation. Building upon the derivation of the cost function and addressing the observability problem through an invariance to both types of transformations, we can now perform nonlinear least squares estimation to optimize our system’s state.

A. Optimization
For simplicity, we give the inputs to our state calculation a simple denotation

\[
Z_k = \{C_i, J_{i,j}\}
\]

Where \( C_i \) is a set of images collected at a frame \( i \) containing all three-dimensional feature points at that frame, \( J_{i,j} \) is a set of IMU measurements from frame \( i \) to frame \( j \), and \( Z_k \) simply contains these two sets of measurements up until frame \( k \). The IMU state \( (R_k, p_k) \in SE(3) \) will be represented as \( \tilde{Z}_k \). It is now possible to represent the maximum a posteriori estimation as

\[
X^* = \arg \max_x p(X|Z_k)
\]

\[
= \arg \max_x \prod_i p(\tilde{Z}_k, C_i) \prod_j p(J_{k,j} | C_i, c_{k,l})
\]

Where \( X = \{Z_k, C_i\} \) includes the observed feature points \( l \) and the IMU states from the initial frame until the \( k \)-th frame while \( c_{k,l} \) are the two-dimensional observed landmarks (which correspond to features) at the \( k \)-th frame. For clarification the extended definition consists of the IMU uncertainty and the feature measurement uncertainty. Since the MAP estimate also corresponds to the minimum of the negative log-posterior, we can formulate the following nonlinear least squares interpretation

\[
X^* = \arg \min_{X} -\log p(X|Z_k)
\]

\[
= \arg \min_{X} \sum_i ||r(\tilde{J}_i, J_{i+1})||^2_0 + \sum_i ||r(\tilde{J}_i, C_i)||_2^2
\]

Where \( r() \) are the respective uncertainties and \( \Sigma_c \) is the covariance in feature measurements.

B. Solving Nonlinear Least Squares
Since the state space of the nonlinear least squares is not part of SE(3), we use the retraction operator in order to solve the algorithm using the Gauss-Newton method. The entire process can be characterized as follows

\[
X = \{e_i, e_l\} = \{\tilde{Z}_k, \oplus e_i, C_i + e_l\}
\]

Where \( e_i \) and \( e_l \) are the Gaussian noise at the \( i \)-th keyframe and reprojective error for all landmarks, respectively as introduced before notation-wise.

IV. EXPERIMENTAL RESULTS
Two different tests were carried out, the first of which was on the EuRoC datasets [20] with a Laptop that had an Intel Core i5-6200U. From the dataset, only readings from the IMU and left image of the stereo camera were used for our algorithm. For the second test, a custom quadrotor was built out of generic parts with an Intel NUC7I7BNH onboard, a Pixhawk 2.4.8 flight controller module with an inexpensive IMU performing visual-inertial synchronization, a Point Grey Firefly-MV global shutter camera, and a Loitir global shutter stereo camera module for taking synchronized stereo images. The stereo module was mounted for the sake of generating ground truth measurements using a convolutional neural network for stereo matching [21]. This was due to a lack of an optical tracking system. After training the network on a relatively small dataset, ground truth measurements were obtained. The system, based on ORB-SLAM’s code, was implemented using C++, Robotic Operating System (ROS), and Ceres Solver [22].

A. EuRoC Dataset
To evaluate our algorithm, we made use of the popular EuRoC datasets to compare our I-EKF VINS performance against two other state-of-the-art monocular VINS, which were monocular OKVIS [4] and VINS-Mono [23] without loop closure. For a visual representation of our algorithm’s performance on these datasets only one figure is included for brevity, which is Figure I. Table 1, however, shows a summary of the root mean square error (RMSE) of translation along all the trajectories for six of the EuRoC datasets. The numbers in bold correspond to the VINS that achieved the best accuracy on the specific dataset sequence.
In our tracking plot, blue trajectories stand for the ground truth while the red trajectories are the optimized estimated trajectories. It is obvious that I-EKF VINS is the most superior algorithm, beaten by a miniscule amount on only one dataset by VINS-MONO, but significantly outperforming all the other algorithms in the other datasets.

### B. Onboard Test

An onboard real-time test was carried out to extract information regarding the robustness and computational efficiency of our method. The front-end extracted all feature points and then calculated the estimated pose, and then the back-end optimized IMU measurements. The combined time it took for the system to perform one iteration for both the front-end and back-end modules was on average 45 milliseconds. The quadrotor platform previously described can be seen in Figure 2.

The trajectory in indoor test with custom quadrotor platform can be seen in Figure 3.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Translation RMSE (Meters)</th>
<th>ORVIS</th>
<th>VINS-MONO</th>
<th>I-EKF VINS</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1_01_easy</td>
<td>0.08587</td>
<td>0.06144</td>
<td><strong>0.05281</strong></td>
<td></td>
</tr>
<tr>
<td>V1_02_medium</td>
<td>0.11920</td>
<td><strong>0.07417</strong></td>
<td>0.07480</td>
<td></td>
</tr>
<tr>
<td>V2_02_medium</td>
<td>0.16892</td>
<td>0.11194</td>
<td><strong>0.04919</strong></td>
<td></td>
</tr>
<tr>
<td>MH_03_medium</td>
<td>0.23951</td>
<td>0.12188</td>
<td><strong>0.09379</strong></td>
<td></td>
</tr>
<tr>
<td>MH_04_difficult</td>
<td>0.25418</td>
<td>0.15192</td>
<td><strong>0.11564</strong></td>
<td></td>
</tr>
<tr>
<td>MH_05_difficult</td>
<td>0.36178</td>
<td>0.31102</td>
<td><strong>0.19218</strong></td>
<td></td>
</tr>
</tbody>
</table>

In our tracking plot, blue trajectories stand for the ground truth while the red trajectories are the optimized estimated trajectories. It is obvious that I-EKF VINS is the most superior algorithm, beaten by a miniscule amount on only one dataset by VINS-MONO, but significantly outperforming all the other algorithms in the other datasets.

## V. Conclusion

In this paper, a novel invariant filtering method was outlined to address the observability problem for both deterministic and stochastic unobservable transformations whereas a general EKF can only address deterministic transformations. This allowed us to keep better track of the rotational properties of our system and in the end significantly improve accuracy. By using full smoothing in our graph, we did not have to marginalize any of the old IMU states, thus further improving the parallel system’s state optimization.

Moreover, the invariance of our filter was the necessary step that made it possible to take full advantage of on-manifold based preintegration. After properly implementing our system in software, experiments revealed that tracking can be computed efficiently and that our method can achieve more accurate results compared to competing state-of-the-art VINS.

### REFERENCES


