Schmidt-EKF-based Visual-Inertial Moving Object Tracking

Kevin Eckenhoff, Patrick Geneva, Nathaniel Merrill, and Guoquan Huang

Abstract— **In this paper we investigate the effect of tightly-coupled estimation on the performance of visual-inertial localization and dynamic object pose tracking. In particular, we show that while a joint estimation system outperforms its decoupled counterpart when given a "proper" model for the target's motion, inconsistent modeling, such as choosing improper levels for the target's propagation noises, can actually lead to a degradation in ego-motion accuracy. To address the realistic scenario where a good prior knowledge of the target's motion model is not available, we design a new system based on the Schmidt-Kalman Filter (SKF), in which target measurements do not update the navigation states, however all correlations are still properly tracked. This allows for both consistent modeling of the target errors and the ability to update target estimates whenever the tracking sensor receives non-target data such as bearing measurements to static, 3D environmental features. We show in extensive simulation that this system, along with a robot-centric representation of the target, leads to robust estimation performance even in the presence of an inconsistent target motion model. Finally, the system is validated in a real-world experiment, and is shown to offer accurate localization and object pose tracking performance.**

I. Introduction

The ability for a sensor-platform to track its ego-motion and perceive its environment is a critical component in many autonomous systems. Cameras and inertial measurement units (IMUs) have become the standard sensor deployment for many of these applications such as unmanned autonomous vehicles (UAVs) and mobile devices [1] due to the affordability and light-weight nature of these sensors. As such, visual-inertial odometry (VIO), which fuses visual-inertial data to estimate the ego-motion of the platform, has seen a recent explosion in research efforts [2]–[6].

In many robotics applications, not only is the user interested in estimating their own motion, but also the tracking of external bodies (targets) whose states are only indirectly observed through exteroceptive sensors mounted on the tracking robot. External object tracking may be necessary for safe navigation in dynamic environments, as in autonomous driving [7], or may even be the overall goal of the sensor deployment, as in military surveillance. For these reasons, the problem of simultaneous localization, mapping, and moving object tracking (SLAMMOT) is important but challenging [8]. However, estimating the poses and motion of external bodies using visual-inertial sensing has received less attention, with a few notable exceptions [9], [10].

When performing SLAMMOT, a question remains open till date whether to use a tightly-coupled or loosely-coupled approach [8]. In the former, the ego-motion and object tracking are formulated as a single joint estimation problem [10], while in the latter the processes are performed separately, often conditioning the target tracking on the output of the localization module [11]. In this paper, building off our prior work on tightly-coupled visual-inertial localization and rigid body target tracking [10], we further investigate the effect that target motion modeling has on overall estimation performance. In particular, we show that while tightly-coupling the systems leads to improved accuracy of both processes (that is, the localization performance is improved when co-estimating the target rather than simply discarding these measurements), this benefit is only seen if the target motion noise values are "properly" chosen. If they are overconfident, this will actually lead to a degrading of the VIO performance. This can be particularly catastrophic in cases where an autonomous vehicle performs the tracking and relies on the accuracy of its VIO to maintain operation.

Therefore, we propose to leverage Schmidt-Kalman Filtering (SKF) [12], an extension to the standard Extended Kalman Filter (EKF), and employ a local (robot-centric) representation of the target, in particular, when there is low confidence in the chosen target model. This allows for the VIO estimates to be provably the same as if target measurements were discarded, however all correlations between the target and IMU are still properly modeled. A summary of this work's contributions are the following:

- We investigate the choice of dynamic object motion noises in a joint visual-inertial localization and target tracking filter, and show through simulation that the selection of these values can lead to either improved or degraded accuracy of the VIO.
- We offer a new Schmidt-EKF formulation that does not allow target measurements to update navigation states while still consistently tracking all correlations, and show that this system can lead to robust estimation accuracy even with inconsistent model selection.
- We advocate a robot-centric representation of the target's pose that is shown to offer improved performance for both the EKF and SKF formulations.
- The proposed system is validated in a real-world experiment where it is shown to offer accurate localization and dynamic object pose tracking estimation.

II. Related Work

While SLAMMOT has seen study in other works [8], [13], few leverage visual-inertial sensors coupled with a pose representation of the target. In particular, many target tracking systems treat the target as a single 3D point [14]. With this, only measurements of this representative feature can be used to update the target, thereby ignoring the additional information provided by other features that typically

This work was partially supported by the University of Delaware (UD) College of Engineering, the NSF (IIS-1924897), the ARL (W911NF-19-2-0226, JWS 10-051-003), Bosch Research, and Google ARCore. Geneva was partially supported by the Delaware Space Grant College and Fellowship Program (NASA Grant NNX15AI19H).

The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. {keck,pgeneva,nmerrill,ghuang}@udel.edu

exist on the object's body. Another limitation of a point particle model is that it requires continuous observation of this single point in order to update. However, in many scenarios the tracking robot may view the target from varied viewing angles thereby occluding this feature despite the object remaining in view of the sensor. Treating the target as a pose with a rigidly connected point cloud relaxes this assumption as we can gain information of the target state when *any* of its features are viewed.

Works that do perform full target pose estimation often leverage more advanced sensors such as LIDAR [15] or RGB-D cameras [16], limiting application for lightweight Micro-Aerial Vehicles (MAVs) due to weight and cost constraints. Other target pose-based estimation methods require additional prior knowledge to handle ambiguity of the estimation problem, such as the target's scale in monocular vision. For example, Li et. al [11] used a dimensional prior of the target and conditioned on the output of the ego-motion estimation (a decoupled approach), and thus did not account for the uncertainty in these estimates.

The work closest to ours, introduced by Qiu et al. [9], utilized a robust monocular visual-inertial batch-based estimator [3] which was first used to track the motion of the sensor platform. The target object was detected using the learning-based object recognition system YOLO [17], and a vision-only structure-from-motion problem was then used to estimate (up-to-scale) the point cloud of the target as well as the relative pose between the camera and target, while metric scale was estimated using trace correlation. While this system was shown to offer robust monocular pose-tracking performance, due to the lack of explicitly estimating the target motion parameters, it is unclear whether this method can be used for active tracking purposes as future target states are not predicted. By contrast, our system fuses observations of the target in a tightly-coupled probabilistic formulation, and is able to improve overall trajectory accuracy through target observation information, its assumed motion model, and proper modeling of its uncertainty. As we additionally estimate motion parameters, we are also able to both predict the target's pose at future timesteps and provide an associated uncertainty of this prediction that can be useful for active tracking scenarios [18].

In our previous work [10], which was built on the light-weight Multi-State Constraint Kalman Filter (MSCKF) framework [19], we advocated the tight-coupling of the tracker's visual-inertial navigation and the target rigid-body pose estimation to improve the accuracy of both processes. We represented the tracking robot and target in the same global frame and analyzed the impact of different assumed target motion models. By contrast, in this paper, we advocate the "local" representation of the target pose that is expressed in the tracking robot's frame of reference. We additionally investigate how in certain scenarios the original tightly-coupled system may lead to a decreased trajectory accuracy in the presence of poor target motion model assumptions. While previous methods handle model uncertainty by using multiple estimators, each using a different model, these lead to a large computational increase [20]. To address this issue, we present an alternative solution that can handle incorrect motion assumptions by leveraging the Schmidt-Kalman Filter [12] formulation, allowing for consistent estimation of the target while preventing corruption of the tracking robot's VIO.

The SKF formulation updates estimates for only a certain subset of variables, keeping the estimates of a set of "nuisance" parameters fixed, while still consistently tracking all correlations. While this leads to computational gains, as has been investigated within the VIO community to reduce complexity [21]–[23], here we leverage the SKF to prevent inconsistent target models from corrupting the trajectory estimates of the tracking robot. We note the SKF has been previously investigated in the context of target tracking to address this issue or to avoid estimating navigation errors, however these works do not consider the visual-inertial domain or object pose-tracking as in this work [24]–[26].

III. Visual-Inertial Estimation

The proposed visual-inertial target tracking system serves as an extension of the standard MSCKF framework [19]. We define the IMU state to be estimated as:

$$\mathbf{x}_I = \begin{bmatrix} {}^I_G\bar{q}^\top & {}^G\mathbf{p}_I^\top & {}^G\mathbf{v}_I^\top & \mathbf{b}_\omega^\top & \mathbf{b}_a^\top \end{bmatrix}^\top \tag{1}$$

where ${}^I_G\bar{q}$ is the JPL-convention quaternion [27] parametrizing the rotation matrix ${}^I_G\mathbf{R}$ that rotates vectors from the gravity-aligned world frame into the local IMU frame. ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$ are the position and velocity of the IMU as expressed in the global frame, and $\mathbf{b}_\omega$ and $\mathbf{b}_a$ are respectively the gyroscope and accelerometer biases that corrupt the corresponding sensors. The vector error state $\tilde{\mathbf{x}}$ is defined by the mapping $\mathbf{x} = \hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}}$ where $\hat{\mathbf{x}}$ is the current best estimate, and the $\boxplus$ operation maps manifold estimates and correction vectors to an updated manifold element [28]. For vectors this operation is standard addition, while for quaternions we utilize the left multiplicative quaternion error [27].

In addition to this evolving inertial state, following the standard MSCKF formulation, we also keep a estimate of the past IMU poses at the last m imaging times. We denote this set of clone states as:

$$\mathbf{x}_{IC} = \begin{bmatrix} {}^{I_{k-1}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{k-1}}^\top & \cdots & {}^{I_{k-m}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{k-m}}^\top \end{bmatrix}^\top$$

The goal of the system is to track the motion of an external object's pose and its local point cloud of rigidly attached features that can be seen and tracked by the sensor's camera. While here we only consider the local velocity model, as in our prior work [10], one could model the target's motion in many different ways. In the chosen model, the estimated target state is expressed as a pose in some reference frame $\{R\}$ and a local linear and angular velocity.

$$\mathbf{x}_T = \begin{bmatrix} {}^T_R\bar{q}^\top & {}^R\mathbf{p}_T^\top & {}^T\mathbf{v}_T^\top & {}^T\boldsymbol{\omega}_T^\top \end{bmatrix}^\top$$

Note that the angular velocity, ${}^T\boldsymbol{\omega}_T$ is that of the target, and is distinct from the angular velocity of the IMU platform. We distinguish between two target pose representations in this work: global and local. In the global representation, the target pose is represented in the same global frame as the IMU ($R = G$), while in the local representation, the target pose is expressed relative to the IMU ($R = I$).

Lastly, assuming the target object moves as a rigid body, we estimate its local point cloud that is made up by a set of

3D features, ${}^{T}\mathbf{p}_{fi}$ that remain static in the target frame. In order to better add these points into our state using delayed initialization [29], we will also maintain a set of target pose clones:

$$\mathbf{x}_{TC} = \begin{bmatrix} {}^{T_{k-1}}_{R_{k-1}}\bar{q}^{\top} & {}^{R_{k-1}}\mathbf{p}_{T_{k-1}}^{\top} & \cdots & {}^{T_{k-m}}_{R_{k-m}}\bar{q}^{\top} & {}^{R_{k-m}}\mathbf{p}_{T_{k-m}}^{\top} \end{bmatrix}^{\top}$$

Note that the target clones in the local formulation are expressed with respect to the corresponding IMU clone, rather than the evolving IMU state. The full state estimated for a single target feature is then given by:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^{\top} & \mathbf{x}_T^{\top} & \mathbf{x}_{IC}^{\top} & \mathbf{x}_{TC}^{\top} & {}^{T}\mathbf{p}_f^{\top} \end{bmatrix}^{\top} \tag{2}$$

A. System Dynamic Models

The visual-inertial system under consideration contains both the IMU (i.e., sensor platform) and the target dynamics. The inertial dynamics is standard and can be found in [27]. Unlike the sensor platform, we are not privy to any direct proprioceptive information for the target, and must perform prediction only using its estimated motion parameters. The dynamics of the global target model are given by:

$${}^{T}_{G}\dot{\mathbf{R}} = -\lfloor {}^{T}\boldsymbol{\omega}_T \times \rfloor {}^{T}_{G}\mathbf{R}, \quad {}^{G}\dot{\mathbf{p}}_T = {}^{G}_{T}\mathbf{R}\,{}^{T}\mathbf{v}_T \tag{3}$$

In the local representation the evolution of the target is coupled with that of the IMU:

$$\begin{aligned} {}^{T}_{I}\dot{\mathbf{R}} &= {}^{T}_{G}\dot{\mathbf{R}}{}^{G}_{I}\mathbf{R} + {}^{T}_{G}\mathbf{R}{}^{G}_{I}\dot{\mathbf{R}} \\ &= -\lfloor {}^{T}\boldsymbol{\omega}_T \times \rfloor {}^{T}_{G}\mathbf{R}{}^{G}_{I}\mathbf{R} + {}^{T}_{G}\mathbf{R}{}^{G}_{I}\mathbf{R}\lfloor {}^{I}\boldsymbol{\omega}_I \times \rfloor \\ &= -\lfloor {}^{T}\boldsymbol{\omega}_T \times \rfloor {}^{T}_{I}\mathbf{R} + {}^{T}_{I}\mathbf{R}\lfloor {}^{I}\boldsymbol{\omega}_I \times \rfloor \end{aligned} \tag{4}$$

$$\begin{aligned} {}^{I}\mathbf{p}_T &= {}^{I}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_T - {}^{G}\mathbf{p}_I\right) \;\Rightarrow \\ {}^{I}\dot{\mathbf{p}}_T &= -\lfloor {}^{I}\boldsymbol{\omega}_I \times \rfloor {}^{I}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_T - {}^{G}\mathbf{p}_I\right) + {}^{I}_{G}\mathbf{R}\left({}^{G}\mathbf{v}_T - {}^{G}\mathbf{v}_I\right) \\ &= -\lfloor {}^{I}\boldsymbol{\omega}_I \times \rfloor {}^{I}\mathbf{p}_T + {}^{I}_{T}\mathbf{R}\,{}^{T}\mathbf{v}_T - {}^{I}_{G}\mathbf{R}\,{}^{G}\mathbf{v}_I \end{aligned} \tag{5}$$

In both representations, we model the local linear and angular velocities as random walks:

$${}^{T}\dot{\mathbf{v}}_T = \mathbf{n}_{vt} \;, {}^{T}\dot{\boldsymbol{\omega}}_T = \mathbf{n}_{\omega t} \tag{6}$$

Thus the amount that the target conforms to the assumed motion model is captured by controlling the values of its propagation noise (6).

1) Covariance Propagation: Based on the above tracking robot and target motion models, we can define the propagation of the error state of the stacked system. Let $\mathbf{x}_S$ denote the set of zero-dynamics static variables (whose true values do not evolve in time, such as static environmental or target features, as well as possible fixed calibration parameters). The overall linearized error state evolution is then given by:

$$\begin{bmatrix} \dot{\tilde{\mathbf{x}}}_I \\ \dot{\tilde{\mathbf{x}}}_T \\ \dot{\tilde{\mathbf{x}}}_S \end{bmatrix} \approx \begin{bmatrix} \mathbf{F}_x & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_x & \mathbf{A}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_I \\ \tilde{\mathbf{x}}_T \\ \tilde{\mathbf{x}}_S \end{bmatrix} + \begin{bmatrix} \mathbf{G}_x & \mathbf{0} \\ \mathbf{\Gamma}_x & \mathbf{\Gamma}_T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{n}_I \\ \mathbf{n}_T \end{bmatrix}$$

where $\mathbf{F}_x$ is the Jacobian of the IMU's error state evolution with respect to the IMU errors, $\mathbf{A}_x$ and $\mathbf{A}_T$ are the Jacobians of the target's error state evolution with respect to the IMU and target errors. Lastly $\mathbf{G}_x, \mathbf{\Gamma}_x$, and $\mathbf{\Gamma}_T$ are the Jacobians with respect to the IMU and target propagation noises ($\mathbf{n}_I$ and $\mathbf{n}_T$). In the case of the global target pose representation, the IMU and target evolution will be decoupled, and thus both $\mathbf{A}_x$ and $\mathbf{\Gamma}_x$ will be zero matrices. From these continuous error-state dynamics, the standard EKF propagation can be performed [27].

B. General Target Update

During the motion of the tracking sensor platform and target, camera bearing measurements to features on the target will be collected. The measurement function is given by:

$$\mathbf{z}_{if} = \mathbf{\Pi}\left({}^{C_i}\mathbf{p}_f\right) + \mathbf{n}_{if} \tag{7}$$

Here ${}^{C_i}\mathbf{p}_f$ denotes the position of the 3D feature in the measuring camera frame, $\mathbf{\Pi}$ is the projection function $\mathbf{\Pi}([x\ y\ z]^{\top}) = [x/z\ y/z]^{\top}$ mapping a feature position expressed in the camera frame into the corresponding normalized pixel coordinate measurement, and $\mathbf{n}_{if} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{if}\right)$ is the Gaussian white-noise corrupting the measurement. The position of the feature in the camera frame using the global and local representations is given respectively by:

$$\begin{aligned} {}^{C_i}\mathbf{p}_f &= {}^{C}_{I}\mathbf{R}{}^{I}_{G}\mathbf{R}\left({}^{G}_{T}\mathbf{R}\,{}^{T}\mathbf{p}_f + {}^{G}\mathbf{p}_T - {}^{G}\mathbf{p}_I\right) + {}^{C}\mathbf{p}_I && (8) \\ &= {}^{C}_{I}\mathbf{R}\left({}^{I}_{T}\mathbf{R}\,{}^{T}\mathbf{p}_f + {}^{I}\mathbf{p}_T\right) + {}^{C}\mathbf{p}_I && (9) \end{aligned}$$

where $\{{}^{C}_{I}\mathbf{R}, {}^{C}\mathbf{p}_f\}$ represents the relative pose between the IMU and camera. After linearizing our target feature measurement (7) about the current state estimate we perform an EKF update of the state estimate, $\hat{\mathbf{x}}$, and covariance $\mathbf{P}$:

$$\mathbf{K} = \mathbf{P}^{\ominus}\mathbf{H}^{\top}\mathbf{S}^{-1} := \mathbf{L}\mathbf{S}^{-1} \tag{10}$$

$$\hat{\mathbf{x}}^{\oplus} = \hat{\mathbf{x}}^{\ominus} \boxplus \mathbf{K}(\mathbf{z}_{if} - \mathbf{\Pi}({}^{C_i}\hat{\mathbf{p}}_f)) \tag{11}$$

$$\mathbf{P}^{\oplus} = \mathbf{P}^{\ominus} - \mathbf{K}\mathbf{H}^{\top}\mathbf{P}^{\ominus} \tag{12}$$

where $\mathbf{H}$ is the measurement Jacobian, $\mathbf{S} = \mathbf{H}\mathbf{P}^{\ominus}\mathbf{H}^{\top} + \mathbf{R}_{if}$, and ${}^{C_i}\hat{\mathbf{p}}_f$ is the result of evaluating evaluating (8) or (9) using the current best state estimates. In addition we use $\ominus$ and $\oplus$ to distinguish between the state/covariance before and after update, respectively. Besides these described target measurements, the system additionally performs updates using bearing measurements to static environmental features in the standard MSCKF manner [19].

IV. Numerical Analysis of Modeling Errors

While it has been shown that the co-estimation of target tracking and localization within a tightly-coupled filter can improve performance [30], we have found through simulations that this conclusion is more nuanced. At the root of the proposed framework is an assumption that the observed target will follow the selected motion model which we have in order to perform prediction of the target's future trajectory. In the case that the target model is not accurate, e.g., too small of noises are used in (6), inconsistencies will be introduced as we are overconfident in how the target evolves and, in the worst case, negatively affect the accuracy of the tracking robot's trajectory. Conversely, if we pick too large of noise values, then the motion parameter estimation becomes unstable, and we do not gain much information to improve the tracking robot's estimate.

To demonstrate this, a numerical simulation scenario was created, where a UAV equipped with an IMU and a stereo camera follows a planar target robot which travels in a semi-circular pattern. A B-spline was fitted to the tracking robot's trajectory to allow for calculation of the groundtruth angular velocities and linear accelerations. From these, the noisy IMU measurements (and corresponding random-walk biases) were simulated at a rate of 200 Hz. A map of 3D points were simulated along the floor and borders of the workspace.

Synthetic stereo images were generated at a frequency of 10 Hz by projecting the map points into the groundtruth camera poses and corrupted with one pixel noise. The target's point cloud was simulated by rigidly attaching a set of points lying on the surface of a one meter cube to the pose of the groundtruth target at each timestep. Both occlusions due to the target and randomly losing track of a given feature were simulated to model problems faced by real-world front-ends. As the target object operated on a plane, we modeled the linear velocity as a random walk only in the local x and y directions, while the angular velocity was a random walk along local z-axis. We note that this type of target is very common in real-world scenarios, such as when tracking a terrestrial vehicle.

To evaluate the effect of different noise parameters on estimation, we chose a single noise standard deviation $\sigma_t = \sigma_{wz} = \sigma_{vx} = \sigma_{vy}$ to drive the *assumed* random walks, see Eq. (6), and performed a parameter sweep over a series of values. Note that for all trials we utilized the same groundtruth IMU and target trajectories, and only varied the noise levels utilized by the filter during target propagation. While a more sophisticated sweep may be performed that varies the noise in all directions, we found that a single parameter was sufficient to demonstrate the discussed issues. For each assumed noise strength, 30 Monte Carlo runs were simulated using the same groundtruth trajectory while each run had a different realization for the IMU and pixel measurement noises. The average Root Mean Squared Error (RMSE) values shown in Table I clearly illustrate the issues caused by inconsistencies when assuming incorrect model noises. The standard VIO without estimating the external target is able to achieve an average RMSE of 0.231 meters and 1.397 degrees for a trajectory of 165 meters. While localization performance improves when using "proper" noise values, the overconfidence introduced by choosing target propagation noises that are too small corrupts the overall system. In addition, we found that having too large of noises may also harm VIO accuracy. This is most likely because at these levels the motion parameter estimation become very unstable leading to poor target predictions that may cause large linearization errors.

Looking at a typical single run and its error bounds, as shown in Figure 1, tightly-coupling of the target tracking reduces the uncertainty compared to stand-alone VIO while remaining consistent (that is, the errors remain in the bounds provided by the covariance) in the case of "good" noise parameters. When the chosen noise is too small, the resulting trajectory error is inconsistent, thereby showing the corruption caused by incorrect modeling of the target motion. Thus, the effectiveness of tightly-coupling the estimation heavily depends on the choice of motion noise parameters. This has profound impact on the application of tightly-coupled target estimation algorithms in the real-world as the choice of correct noise values is difficult and can leave the system vulnerable to inaccuracies and in the worse case may even cause filter divergence. This motivates the proposed Schmidt-EKF formulation that can allow for modeling of the uncertainties between the tracking robot's states and the target's without the inconsistencies of the target estimates affecting the tracking robot's trajectory.

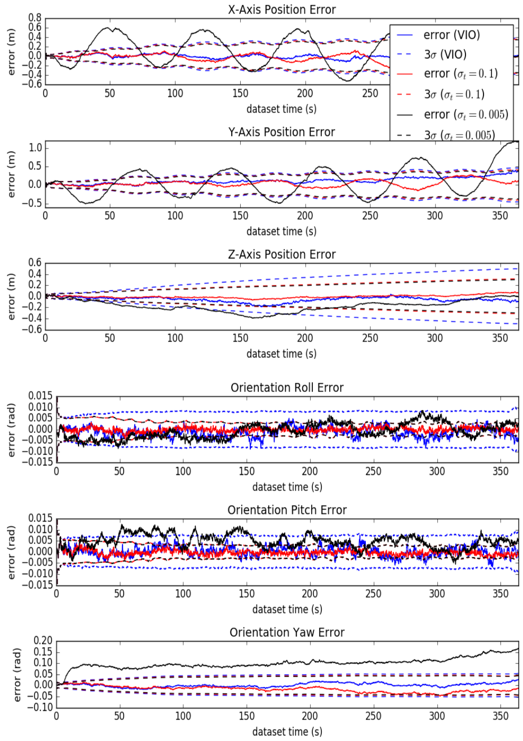

Fig. 1: Estimation errors of IMU global pose when using VIO by itself, tightly-coupled tracking with a "good" target $\sigma_t = 0.1$, as well as an overconfident $\sigma_t = 0.005$. In the tightly-coupled system with "proper" noises, the error bounds are decreased and the estimate remains consistent, showing that the fusion of target information has improved localization performance. For an overconfident noise, the estimator becomes inconsistent.

V. Schmidt EKF Update

In the case that we want to prevent corruption of the tracking robot's estimates, we leverage the SKF [12] to update the target states; that is, when processing measurements to the target, we "schmidt" the inertial state (and corresponding clones) and thus only update the target estimates and their correlation with the tracking robot's states. More explicitly, we partition the state into two, the tracking robot's states $\mathbf{x}_R$ and target parameters, $\mathbf{x}_T$, such that $\mathbf{x} = [\mathbf{x}_R^\top \ \mathbf{x}_T^\top]^\top$, and decompose the covariance as: $\mathbf{P} = \begin{bmatrix}\mathbf{P}_{RR} & \mathbf{P}_{RT}\\ \mathbf{P}_{TR} & \mathbf{P}_{TT}\end{bmatrix}$. Now, let us consider a measurement residual $\mathbf{r}$ that is being used to update the state. The SKF does not update the tracking robot's states by setting their Kalman gain to zero (i.e. $\mathbf{K} = [\mathbf{K}_R^\top \ \mathbf{K}_T^\top]^\top \to [\mathbf{0}^\top \ \mathbf{K}_T^\top]^\top$). The update equations become the following:

$$\hat{\mathbf{x}}^{\oplus} = \hat{\mathbf{x}}^{\ominus} \boxplus \begin{bmatrix}\mathbf{0}\\ \mathbf{K}_T\end{bmatrix}\mathbf{r} = \hat{\mathbf{x}}^{\ominus} \boxplus \begin{bmatrix}\mathbf{0}\\ \mathbf{L}_T\end{bmatrix}\mathbf{S}^{-1}\mathbf{r} \tag{13}$$

$$\mathbf{P}^{\oplus} = \mathbf{P}^{\ominus} - \begin{bmatrix}\mathbf{0} & \mathbf{L}_R\mathbf{S}^{-1}\mathbf{L}_T^\top\\ \mathbf{L}_T\mathbf{S}^{-1}\mathbf{L}_R^\top & \mathbf{L}_T\mathbf{S}^{-1}\mathbf{L}_T^\top\end{bmatrix} \tag{14}$$

where $\begin{bmatrix}\mathbf{L}_R^\top & \mathbf{L}_T^\top\end{bmatrix}^\top = \mathbf{L}$ from (10). Clearly, the tracking robot states $\mathbf{x}_R$ and marginal covariance $\mathbf{P}_{RR}$ do not change in this update, preventing the corruption of these estimates in the case that the target estimate is inconsistent. Therefore

TABLE I: Tracking robot and target average RMSE in meters / degrees of the global (G) and local (L) tightly-coupled estimator for different values of σ_t. We evaluate the target pose in the tracking robot's IMU frame ${}^T_I\mathbf{T}$, the target pose in the global frame, ${}^T_G\mathbf{T}$, and the IMU pose in the global frame ${}^I_G\mathbf{T}$.

σ_t	${}^T_I\mathbf{T}$ (G)	${}^T_G\mathbf{T}$ (G)	${}^I_G\mathbf{T}$ (G)	${}^T_I\mathbf{T}$ (L)	${}^T_G\mathbf{T}$ (L)	${}^I_G\mathbf{T}$ (L)
0.001	0.01172 / 1.069	2.149 / 12.843	2.030 / 12.948	0.01177 / 0.948	1.883 / 8.944	1.839 / 8.687
0.005	0.00720 / 0.545	0.277 / 1.519	0.271 / 1.678	0.00746 / 0.543	0.259 / 1.547	0.254 / 1.566
0.010	0.00697 / 0.499	0.241 / 1.525	0.236 / 1.690	0.00724 / 0.497	0.220 / 1.322	0.217 / 1.388
0.050	0.00646 / 0.319	0.211 / 1.478	0.207 / 1.504	0.00664 / 0.315	0.203 / 1.313	0.199 / 1.341
0.100	0.00657 / 0.339	0.211 / 1.500	0.207 / 1.502	0.00661 / 0.340	0.197 / 1.291	0.194 / 1.293
0.500	0.00835 / 0.687	0.284 / 1.719	0.279 / 1.779	0.00848 / 0.627	0.235 / 1.370	0.232 / 1.378

TABLE II: Tracking robot and target average RMSE for the proposed SKF formulation. For very overconfident noise values, the global representation causes target estimate divergence.

σ_t	${}^T_I\mathbf{T}$ (G)	${}^T_G\mathbf{T}$ (G)	${}^I_G\mathbf{T}$ (G)	${}^T_I\mathbf{T}$ (L)	${}^T_G\mathbf{T}$ (L)	${}^I_G\mathbf{T}$ (L)
0.001	×	×	0.231 / 1.397	0.01092 / 0.879	0.234 / 1.720	0.231 / 1.397
0.005	×	×	0.231 / 1.397	0.00763 / 0.528	0.234 / 1.524	0.231 / 1.397
0.010	0.0408 / 0.555	0.231 / 1.472	0.231 / 1.397	0.00732 / 0.488	0.234 / 1.511	0.231 / 1.397
0.050	0.0410 / 0.411	0.229 / 1.393	0.231 / 1.397	0.00665 / 0.334	0.234 / 1.399	0.231 / 1.397
0.100	0.0432 / 0.430	0.229 / 1.390	0.231 / 1.397	0.00664 / 0.364	0.234 / 1.413	0.231 / 1.397
0.500	0.0623 / 0.744	0.233 / 1.633	0.231 / 1.397	0.00853 / 0.639	0.234 / 1.617	0.231 / 1.397

the resulting VIO estimates are equivalent to if the target measurements had been discarded. It is important to note that the SKF still tracks correlations in the covariance that can be shown to be a conservative approximation of the EKF update [21]. We also highlight that the SKF is only used when updating using target measurements. During processing of the visual data corresponding to the static scene, the full state is updated normally. This allows corrections to the IMU to correct the target estimates, which is only possible because of the consistently tracked correlations.

Due to the fact that the SKF does not update a portion of the state, the representation of the estimated variables becomes crucial to the performance [24]. In the global model, the filter is free to fully update the global pose of the target, while in the local model, it fully updates the relative pose between the IMU and target. Intuitively, as target measurements are relative to the sensing platform, it makes more sense to fully update the relative relationship between the two. To investigate the effect of this robot-centric representation, we reran the noise sweep of the previous section using the local formulation, as shown in Table I. These results illustrate that for the tightly-coupled system the robot-centric filter outperforms its global counterpart in terms of both target and VIO accuracy. This is most likely due to the fact that linearization errors tend to be much smaller in the local frame. However, this system still displayed the same problems when using a poor motion model.

We tested the proposed SKF formulation with both the global and local representations (Table II). As expected, the systems cannot benefit from the tightly-coupled estimation for the “proper” noise values, while the VIO cannot be corrupted even when the target model is inconsistent. In addition, we experimentally found that the global target pose filter was more prone to target estimate divergence at the lower noise levels (typically following large target updates), while the robot-centric method was always able to provide accurate tracking performance, even when using an inconsistent motion model. This motivates our choice of a local representation for both the EKF and SKF formulations, and shows that our local SKF filter offers robustness to poor target models.

VI. Extension to Real-World Tracking

We next evaluated the proposed local SKF-based localization and object tracking in a real-world scenario using a stereo visual-inertial rig. In what follows we first discuss the visual feature front-end needed to both segment the target and then track features on that target over its trajectory. From a high level, we performed segmentation on incoming images using a variation of UNet [31] that generates a mask of the target for each image. Using this mask, we then performed separate visual tracking for environmental and target features which can then be fused in the proposed estimator.

A. Deep Learning-based Target Detection

Due to the absence of an annotated visual-inertial target tracking dataset with groundtruth trajectories for the robot and target, we opted to collect and annotate our own dataset tracking a large remote controlled car. We collected a training, validation and testing dataset from the left camera at 1Hz, resulting in 279 images for training and validation (with 10 percent randomly chosen for validation), and 85 for testing. These images were labeled by hand to create the groundtruth masks. The camera used to collect the images had a fisheye lense, but the masks were generated without undistoring the images to avoid having to redistort later on. However, due to this, we only used random left-right flipping to augment the data and not rotations and random cropping.

A variant of the popular UNet architecture was used to segment the target. In our test, we used sub-pixel convolution [32] in place of the original upsample layer proposed for efficiency. Additionally, we replaced the ReLU activation with ELU [33] to avoid the need for batch normalization – again, for efficiency. We also used zero padding to avoid the need for cropping the features used in skip connections. The network can predict on average in less than 20 milliseconds on a GTX 1080Ti graphics card using a 320×240 resolution. To test the network, we used the mean intersection over union (mIOU) and pixel-wise accuracy metrics. Pixel-wise

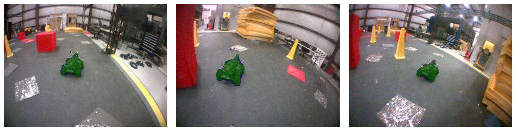

Fig. 2: Visual results of the network on the testing dataset. True positive pixels have been marked green, false positives blue, and false negatives pink.

accuracy in our case is defined by $\frac{TP+TN}{TP+TN+FP+FN}$, where TP, TN, FP, and FN are the true positive, true negative, false positive, and false negative predictions, respectively. The network achieved an mIOU of 0.628 and a pixel-wise accuracy of 99.1% on the testing dataset (see Fig. 2).

Since the target typically occupies a small portion of the overall pixels, we opted to extract target features from the bounding box of the segmentation mask instead of the entire image, while environmental features were extracted from the full image. To do this, we first removed the majority of false positives from the mask through a series of erosion and dilation operations. The bounding box was then taken as the rectangle about the blob with the largest area. A separate KF was used to track the bounding box, thereby smoothing the noisy masks.

B. Visual Feature Tracking

We utilized FAST detection [34] across multiple grids of the image to ensure that we achieved a uniform distribution of features. Only features that fell within the mask were labeled as target features. We performed KLT tracking [35] through the implementation available in OpenCV [36] for environmental features, and ORB descriptor tracking [37] for target features. This tracking was done both from the left-to-right images at the same timestamp as well as left-to-left and right-to-right tracking from the previous images. Outliers were rejected using 8-point RANSAC which was performed independently for the static and target features.

Target features that were tracked in this way for a certain number of frames were initialized as permanent object features. In addition, we performed ORB descriptor matching to the currently estimated target point cloud in order to determine whether new tracks corresponded to previously seen features. We have found that these target loop closing events are vital to the performance of the estimator as they greatly limit the target's drift relative to the IMU (and therefore globally). As mismatched features are common in real-world experiments, we utilized Mahalanobis distance tests to reject inconsistent measurements.

C. Experimental Results

In this experiment, a hand-held visual-inertial rig tracked a remote-controlled car. The network was trained to detect the vehicle using a separately collected dataset in the manner described previously. Note that different from the state of art [9], we do not require a high-resolution camera (which could certainly lead to higher accuracy). We allowed for a maximum of 150 features to be tracked from the point cloud. To limit visual-inertial drift, we added up to 10 SLAM features which were generated from visual tracks that reached the length of the window size (10 images in this test) [29], and were marginalized whenever they were lost. We also performed online estimation of the IMU-camera extrinsincs to handle the possibly poor prior calibration. As this was a (mostly) planar scenario, we chose the noises driving the local z-axis linear velocity and the pitch/roll angular velocities to be zero, as in the simulated example. For this experiment, images were collected at 10 Hz and the entire system ran in real time on a commercial laptop.

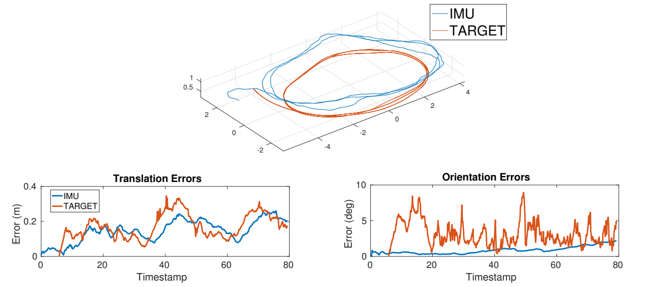

Fig. 3: True trajectory and estimation errors of IMU and target global poses for the proposed SKF system. The total IMU and target trajectories were 62 meters and 52 meters, respectively.

In order to evaluate the accuracy, the tracking scenario was performed in a large Vicon room, thus providing highly accurate pose estimates of both the sensor platform and the target. A purposefully overconfident target motion model was selected to illustrate the proposed method's robustness. Over 30 Monte Carlo runs, the SKF based target tracking had a VIO RMSE of 0.153 m / 1.091 degrees, while the global target errors were 0.183 m / 3.443 degrees. The error-vs-time plots for an example run are shown in Fig 3. By contrast, the tightly-coupled system achieved 0.409 m / 1.640 degrees for the IMU and 0.492 m / 3.037 degrees for the target, thus showing the degradation of the VIO due to the poor motion model. These results indicate that the proposed system is able to accurately estimate both the target and navigation states even when using an overconfident target model. In addition, the mask provided by the network failed for several sequential frames multiple times throughout the test, which our system was able to handle due to the estimation of the target motion parameters.

VII. Conclusions

In this work we have addressed the issue of inconsistent target motion modeling in a tightly-coupled visual-inertial localization and 3D rigid-body target tracking framework. We have shown that when the model accurately describes the target motion, tightly-coupled estimation leads to improved accuracy of both the localization and tracking tasks. We have also illustrated the dangers inherent to overconfident model selection, and have shown that using an SKF approach prevents inconsistent motion noises from corrupting the VIO task while still properly tracking all correlations between the target and navigation states. In addition, we have advocated a local representation of the target for improved estimation performance in both the EKF and SKF formulations. The proposed approach was validated in a real-world tracking scenario and shown to offer accurate estimation of both the ego and target motions even without utilizing a high-resolution camera or continuous successful visual tracking.

REFERENCES

[1] K. J. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis, “A square root inverse filter for efficient vision-aided inertial navigation on mobile devices,” in *Robotics: Science and Systems Conference (RSS)*, 2015.

[2] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, “Vision-aided inertial navigation for spacecraft entry, descent, and landing,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 264 –280, 2009.

[3] 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. 1004–1020, 2018.

[4] 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.

[5] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual-inertial odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[6] G. Huang, “Visual-inertial navigation: A concise review,” in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[7] R. Wang, M. Schwörer, and D. Cremers, “Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras,” in *International Conference on Computer Vision (ICCV), Venice, Italy*, 2017.

[8] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, “Simultaneous localization, mapping and moving object tracking,” *Int. J. Rob. Res.*, vol. 26, no. 9, pp. 889–916, Sept. 2007.

[9] K. Qiu, T. Qin, W. Gao, and S. Shen, “Tracking 3-d motion of dynamic objects using monocular visual-inertial sensing,” *IEEE Transactions on Robotics*, vol. PP, pp. 1–18, 05 2019.

[10] K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, “Tightly-coupled visual-inertial localization and 3D rigid-body target tracking,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 2, pp. 1541–1548, 2019.

[11] P. Li, T. Qin, *et al.*, “Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 646–661.

[12] S. F. Schmidt, “Application of state-space methods to navigation problems,” ser. Advances in Control Systems, C. LEONDES, Ed. Elsevier, 1966, vol. 3, pp. 293 – 340.

[13] J. S. Ortega, “Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach,” *École Doctorale Systèmes, Docteur de lInstitut National Politechnique de Toulouse, PhD theses*, 2007.

[14] M. Chojnacki and V. Indelman, “Vision-based dynamic target trajectory and ego-motion estimation using incremental light bundle adjustment,” *International Journal of Micro Air Vehicles*, vol. 10, no. 2, pp. 157–170, 2018.

[15] A. Azim and O. Aycard, “Detection, classification and tracking of moving objects in a 3d environment,” in *2012 IEEE Intelligent Vehicles Symposium*, June 2012, pp. 802–807.

[16] A. Aldoma, F. Tombari, J. Prankl, A. Richtsfeld, L. Di Stefano, and M. Vincze, “Multimodal cue integration through hypotheses verification for rgb-d object recognition and 6dof pose estimation,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 2104–2111.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[18] H. Wei, W. Lu, P. Zhu, G. Huang, J. Leonard, and S. Ferrari, “Visibility-based motion planning for active target tracking and localization,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, Sept. 14-18 2014, pp. 76–82.

[19] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint Kalman filter for vision-aided inertial navigation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10–14, 2007, pp. 3565–3572.

[20] X. Rong Li and V. P. Jilkov, “Survey of maneuvering target tracking. part v. multiple-model methods,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, Oct 2005.

[21] R. C. DuToit, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis, “Consistent map-based 3d localization on mobile devices,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 6253–6260.

[22] P. Geneva, K. Eckenhoff, and G. Huang, “A linear-complexity EKF for visual-inertial navigation with loop closures,” in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[23] P. Geneva, J. Maley, and G. Huang, “An efficient schmidt-ekf for 3D visual-inertial SLAM,” in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, June 2019, (accepted).

[24] E. F. Brekke and E. F. Wilthil, “Suboptimal kalman filters for target tracking with navigation uncertainty in one dimension,” in *2017 IEEE Aerospace Conference*, March 2017, pp. 1–11.

[25] R. Y. Novoselov, S. M. Herman, S. M. Gadaleta, and A. B. Poore, “Mitigating the effects of residual biases with schmidt-kalman filtering,” in *2005 7th International Conference on Information Fusion*, vol. 1, July 2005, pp. 8 pp.–.

[26] C. Yang, E. Blasch, and P. Douville, “Design of schmidt-kalman filter for target tracking with navigation errors,” in *2010 IEEE Aerospace Conference*, March 2010, pp. 1–12.

[27] N. Trawny and S. I. Roumeliotis, “Indirect Kalman filter for 3D attitude estimation,” University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005.

[28] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, “Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds,” *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.

[29] M. Li, “Visual-inertial odometry on resource-constrained systems,” Ph.D. dissertation, UC Riverside, 2014.

[30] F. M. Mirzaei and S. I. Roumeliotis, “A Kalman filter-based algorithm for IMU-camera calibration,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, Oct. 29 - Nov. 2 2007, pp. 2427–2434.

[31] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015.

[32] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” *CoRR*, vol. abs/1609.05158, 2016.

[33] D. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *CoRR*, vol. abs/1511.07289, 2015.

[34] E. Rosten, R. B. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, 2010.

[35] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. of the International Joint Conference on Artificial Intelligence*, Vancouver, BC, 1981, pp. 674–679.

[36] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[37] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.