Map-based Visual-Inertial Localization: A Numerical Study

Patrick Geneva - pgeneva@udel.edu
Guoquan Huang - ghuang@udel.edu

Department of Mechanical Engineering
University of Delaware, Delaware, USA

RPNG

Robot Perception and Navigation Group (RPNG)
Tech Report - RPNG-2022-MAPPING
Last Updated - Febuary 17, 2022

Contents

1 Modeling of Prior Feature Map Covariance

1.1 Perturbation Methods

We now look at two different techniques for modeling the uncertainty on the 3D feature state: (1) as a Cartesian feature with independent axes, and (2) locally uncertain along the feature bearing and depth in an observing keyframe. The first is representative of a prior map generated from a LiDAR or detailed offline optimization method, while the second is representative of a the output of a bundle adjustment system which optimizes features in an anchored frame of reference and can recover the covariance [1, 2, 3, 4]. In both cases our lightweight map-based filter frontend will leverage the estimates of the features in the global frame (see the presented paper for details [5]).

$$\mathbf{x}_M = \begin{bmatrix} {}^G\mathbf{p}_{f_1}^\top & \cdots & {}^G\mathbf{p}_{f_m}^\top \end{bmatrix}^\top \tag{1}$$

The initial covariance of the system is dependent on how we simulate the perturbations to the initial states. In the case of the Cartesian feature perturbations we have:

$${}^G\mathbf{p}_{f_i,init} = {}^G\mathbf{p}_{f_i} + \mathbf{n}_{xyz} \tag{2}$$

$$\mathbf{n}_{xyz} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{xyz}) \tag{3}$$

$$\mathbf{R}_{xyz} = \sigma_{xyz}^2 \mathbf{I} \tag{4}$$

When simulating, each axis of the true feature gets perturbed by a sample from the Gaussian distribution σ_{xyz}. The estimator is provided with the initial perturbed state ${}^G\mathbf{p}_{f_i,init}$ and prior covariance $\mathbf{R}_{xyz}$.

To recover the perturbed global feature state and covariance for the anchored feature perturbation model, first the anchor frame is determined by finding the first observing keyframe from which the feature was seen from. The feature's true bearing and depth are then perturbed in this frame. Specifically we can calculate the true bearing and depth of a feature in the anchor as:

$${}^A\mathbf{p}_{f_i} = {}^A_G\mathbf{R}({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_A) \tag{5}$$

$$\boldsymbol{\lambda}_i = \begin{bmatrix} \theta_i \\ \phi_i \\ d_i \end{bmatrix} = \begin{bmatrix} \text{atan2}({}^A\mathbf{p}_{f_i,y}, {}^A\mathbf{p}_{f_i,x}) \\ \text{acos}({}^A\mathbf{p}_{f_i,z}/||{}^A\mathbf{p}_{f_i}||) \\ ||{}^A\mathbf{p}_{f_i}|| \end{bmatrix} \tag{6}$$

The feature can then be perturbed as follows:

$$\boldsymbol{\lambda}_{i,init} = \boldsymbol{\lambda}_i + \mathbf{n}_\lambda \tag{7}$$

$$\mathbf{n}_\lambda \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_\lambda) \tag{8}$$

$$\mathbf{R}_\lambda = \begin{bmatrix} \sigma_b^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{bmatrix} \tag{9}$$

where we have perturbed the true feature's bearing and depth in the anchor frame recovering the state $\boldsymbol{\lambda}_{i,init}$ with the Gaussian distributions σ_b and σ_d, respectively. From here, we wish to recover the position of the feature in the global and the uncertainty of the feature. The mean of the feature can be recovered through transforming the feature in the global, while the uncertainty can be found by "propagating" the local feature uncertainty into the global.

$${}^A\mathbf{p}_{f_i,init} = d_{i,init} \begin{bmatrix} \cos(\theta_{i,init})\ \sin(\phi_{i,init}) \\ \sin(\theta_{i,init})\ \sin(\phi_{i,init}) \\ \cos(\phi_{i,init}) \end{bmatrix} \tag{10}$$

$${}^G\mathbf{p}_{f_i,init} = {}^A_G\mathbf{R}^\top {}^A\mathbf{p}_{f_i,init} + {}^G\mathbf{p}_A \tag{11}$$

For the covariance we have the following:

$$\mathbf{R}_{global} = \mathbf{H}\mathbf{R}_{\lambda}\mathbf{H}^{\top} \tag{12}$$

$$\mathbf{H} = {}^{A}_{G}\mathbf{R}^{\top} \begin{bmatrix} -d_i \sin(\theta_i) \sin(\phi_i) & d_i \cos(\theta_i) \cos(\phi_i) & \cos(\theta_i) \sin(\phi_i) \\ d_i \cos(\theta_i) \sin(\phi_i) & d_i \sin(\theta_i) \cos(\phi_i) & \sin(\theta_i) \sin(\phi_i) \\ 0 & -d_i \sin(\phi_i) & \cos(\phi_i) \end{bmatrix} \tag{13}$$

The estimator is provided with the initial perturbed state ${}^{G}\mathbf{p}_{f_i,init}$ and prior covariance $\mathbf{R}_{global}$.

1.2 Extended Simulation Results

Using the 1.2km hand-held Room trajectory we run a series of simulations with the proposed different prior map perturbation methods. The average results have been tabulated in Table 1. First, we can see that all the prior map methods are able to outperform the odometry VIO method. Additionally, even at large noise levels of 12cm-24cm, both the landmark and keyframe methods are still able to gain in both the orientation and position accuracy. The 2D-to-2D method is able to improve the orientation for all noise levels while the localization position estimates can be degraded at the higher noise level of 0.5m keyframe position errors. Additionally, we can see that the 2D-to-3D methods greatly outperform the 2D-to-2D method even at extremely large noise perturbations. This makes sense since the 2D-to-2D indirectly constrain the current pose of the system through additional feature observations, while the 2D-to-3D directly constrain *all* observations for a feature. There seems to be little performance difference between the different 3D feature map noise prior perturbation methods.

It is also interesting to note that while the EKF methods have very good levels of accuracy, the NEES increases with noise perturbation levels. We conjecture this is due to the use of First-estimates Jacobians (FEJ) [6, 7], which can introduce linearization errors at high noise levels (the SKF hides this due to its naturally conservative covariance for the middle noise perturbation levels). This has been investigated in detail in the recently published work by Chen et al. [8]. This is more clear at the larger errors were the linearization errors have become even more hurtful and can actually degrade estimator performance worst than VIO.

Table 1: Average Absolute Trajectory Error (ATE) and Normalized Estimation Error Squared (NEES) over 5 Room dataset runs for different map priors and algorithms. ATE units are in degrees and meters, with NEES being ideally around 3 in magnitude.

	Prior	Algo.	ATE (deg/m)	NEES (ori/pos)	Prior	Algo.	ATE (deg/m)	NEES (ori/pos)	Prior	Algo.	ATE (deg/m)	NEES (ori/pos)
VIO	-	-	2.379 / 0.266	3.518 / 1.592	-	-	2.379 / 0.266	3.518 / 1.592	-	-	2.379 / 0.266	3.518 / 1.592
2D-to-2D	0.5°, 3cm	EKF	0.322 / 0.090	2.926 / 3.333	2D-to-3D Cartesian 3cm	EKF	0.050 / 0.010	5.958 / 6.574	2D-to-3D Bearing 0.1°, 6cm	EKF	0.036 / 0.008	3.785 / 3.973
		SKF	0.370 / 0.098	2.752 / 3.250		SKF	0.064 / 0.021	2.905 / 3.190		SKF	0.058 / 0.016	3.299 / 3.288
	1.0°, 6cm	EKF	0.424 / 0.104	3.226 / 3.703	6cm	EKF	0.066 / 0.014	8.192 / 9.267	0.5°, 12cm	EKF	0.064 / 0.013	7.827 / 8.607
		SKF	0.504 / 0.128	2.802 / 3.471		SKF	0.087 / 0.036	2.862 / 3.212		SKF	0.107 / 0.047	3.235 / 3.302
	3.0°, 12cm	EKF	0.603 / 0.127	4.346 / 5.338	12cm	EKF	0.076 / 0.015	9.286 / 9.448	1.0°, 24cm	EKF	0.067 / 0.014	7.425 / 8.382
		SKF	0.929 / 0.166	3.009 / 3.595		SKF	0.122 / 0.065	2.757 / 3.175		SKF	0.165 / 0.086	3.305 / 3.363
	6.0°, 24cm	EKF	0.876 / 0.214	5.498 / 11.227	24cm	EKF	0.088 / 0.016	11.016 / 9.337	2.0°, 50cm	EKF	0.067 / 0.021	6.741 / 14.320
		SKF	1.442 / 0.249	3.595 / 5.970		SKF	0.191 / 0.121	2.778 / 3.253		SKF	0.270 / 0.164	3.396 / 3.633
	9.0°, 50cm	EKF	2.005 / 0.331	13.771 / 17.988	50cm	EKF	0.114 / 0.027	17.532 / 21.369	3.0°, 1m	EKF	0.107 / 0.031	10.761 / 21.785
		SKF	1.767 / 0.345	3.455 / 6.254		SKF	0.344 / 0.257	3.008 / 3.845		SKF	0.403 / 0.271	3.637 / 4.143

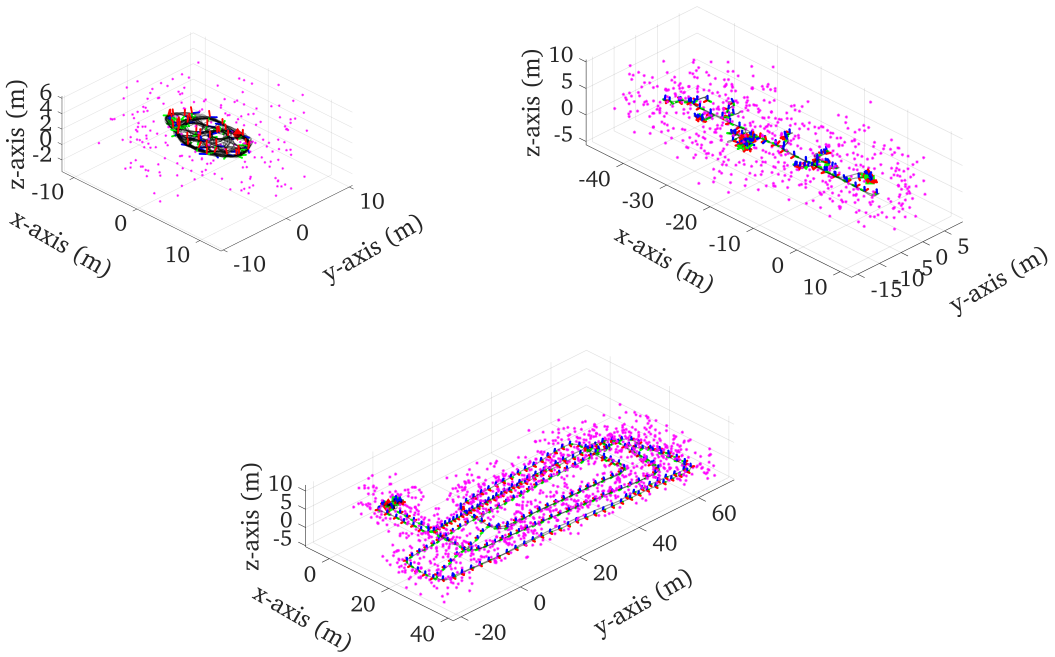

Figure 1: Simulated trajectories of a 1.2km hand-held Room (top left), 300m TUM-VI Corridor (top right), and 925m TUM-VI Magistrale (bottom) trajectory, axes are in units of meters. Every other keyframe is shown to increase clarity with feature depths (purple) being generated between 5 and 7 meters.

2 Investigation of Trajectory Sensitivity

A natural question is how sensitive is the analysis to the selected trajectory. To try to address this concern, we simulated a series of additional datasets and compared them to the one used in the presented paper [5]. We consider two additional simulation trajectories generated from the TUM-VI dataset [9] which were generated using a visual-inertial odometry system (see Figure 1). The camera calibration parameters and IMU intrinsics from the TUM-VI dataset were used for the below experiments. The first additional trajectory is a single floor 300m long trajectory in an office environment generated from the Corridor 1 dataset, while the second is a much larger scale three floor 925m long trajectory generated from the Magistrale 1 dataset. These contrast the single room small scale, but long-term, Room dataset which is 1.2km in length.

2.1 Prior Map Generation

The prior feature and keyframe map is generated by starting at the beginning of the trajectory and moving the camera forward in time at a rate of 4 Hz. At each timestep we project the current landmark map into the camera frame and if the number of seen features falls below our average feature tracking amount we generate new features. If it does, then we generate a feature with a random bearing and depth between 5 and 7 meters. This is repeated until the end of the trajectory is reached and our prior landmark map is complete after applying perturbations. To generate the keyframe map, we repeat this procedure. Specifically at each timestep the current camera must be near an existing keyframe and share a sufficient percentage of common overlapping features; otherwise a new keyframe is created. On failure then a new keyframe is created at this timestep. After generating our keyframes, we project the landmark map into each to generate bearing observations, and both the keyframe poses and observations are perturbed. This generation

logic recreates the similar procedure as offline prior map creation which also requires processes to bound the number of landmarks and keyframes (see [1, 2, 3, 4] for an examples).

2.2 Discussion on Computational Complexity

The resulting state size for each trajectory can be seen in Figure 2. It is clear that the trend of linear state size in terms of the number of tracked features for the 3D feature state and a constant state size for keyframe-based methods remains. Additionally, for the larger scale environments the states, in general, explode in size resulting in a dramatic increases in the computational cost. Figure 3 shows the average time for each update and propagation for a run on each dataset. In general, the EKF takes the most time, the SKF second, and the inflation methods all around the same order. The 2D-to-2D (KF) methods have near constant offset from the VIO time as the number of average features only marginally increases the computational cost due to more measurements. This is a clear advantage when the number of tracked features is large. The 2D-to-3D (PTS) method quickly increases an order of magnitude slower than VIO, which is expected as the state size dramatically grows (see Figure 2). The inflation methods (INF) for both landmark and keyframe prior maps perform as efficiently as VIO due to their near constant run-time and constant state vector size.

Comparing the Room and TUM-VI Corridor timings in Figure 3, the computational cost of EKF-based methods jump in an order for most methods. We can also see that the SKF methods have a very marginal increase in computational cost which is expected as it should only grow linear with the increase in state size (compared to the quadratic order for the EKF). Finally, in the TUM-VI Magistrale dataset, all the EKF-based methods are non-realtime with the SKF keyframe methods being around 50Hz and the point-based SKF method being around 10Hz at the larger feature count size. It is also clear that the keyframe-based methods have a much larger increase in size due to the larger volume they must cover. As compared to the Room dataset, keyframes are unlikely to cover a previously seen area, and thus occur at a higher frequency in the prior map increasing the state size. This shows the advantage of the inflation methods for these larger environments were there is limited to no re-visiting of previously explored areas.

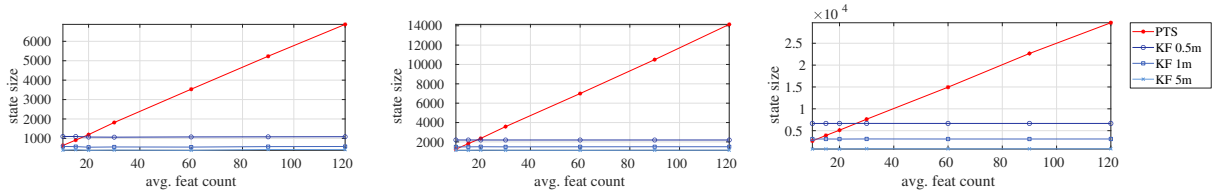

Figure 2: Relation between state state size (number of variables) and the average number of features observed for both landmark-based (PTS) and keyframe-based (KF) maps in the Room (left), TUM-VI Corridor (middle), and TUM-VI Magistrale (right) datasets. Different maximum keyframe distance thresholds are also plotted.

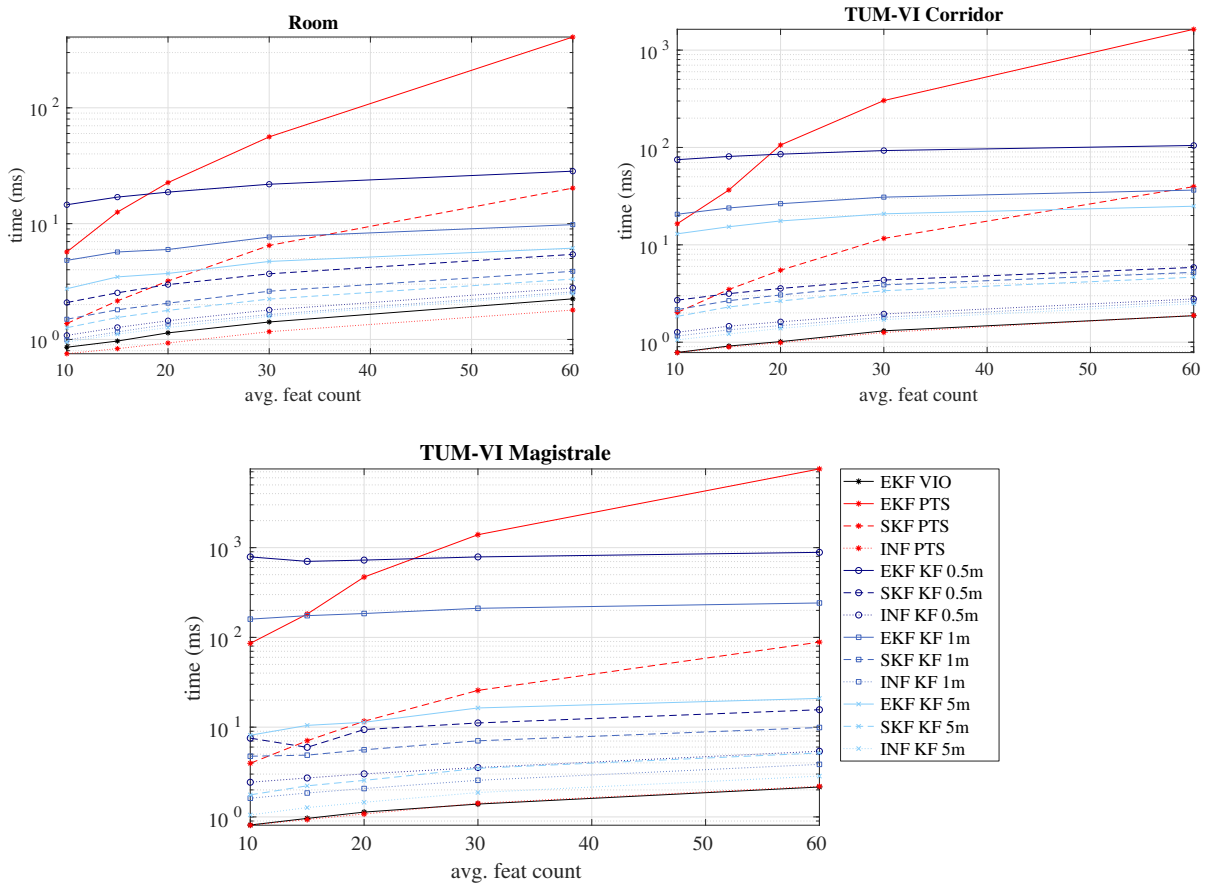

Figure 3: Runtime in milliseconds for both propagation and update without (VIO) and with both landmark-based (PTS) and keyframe-based (KF) maps for the Room (top left), TUM-VI Corridor (top right), and TUM-VI Magistrale (bottom) datasets. Keyframe-based map are reported for different max keyframe distances.

References

[1] Thomas Schneider, Marcin Dymczyk, Marius Fehr, Kevin Egger, Simon Lynen, Igor Gilitschenski, and Roland Siegwart. "maplab: An open framework for research in visual-inertial mapping and localization". In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1418–1425.

[2] Marcin Dymczyk, Simon Lynen, Michael Bosse, and Roland Siegwart. "Keep it brief: Scalable creation of compressed localization maps". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 2536–2542.

[3] Marcin Dymczyk, Simon Lynen, Titus Cieslewski, Michael Bosse, Roland Siegwart, and Paul Furgale. "The gist of maps-summarizing experience for lifelong localization". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 2767–2773.

[4] Chao X. Guo, Kourosh Sartipi, Ryan C. DuToit, Georgios A. Georgiou, Ruipeng Li, John O'Leary, Esha D. Nerurkar, Joel A. Hesch, and Stergios I. Roumeliotis. "Resource-Aware Large-Scale Cooperative Three-Dimensional Mapping Using Multiple Mobile Devices". In: *IEEE Transactions on Robotics* 34.5 (2018), pp. 1349–1369.

[5] Patrick Geneva and Guoquan Huang. "Map-based Visual-Inertial Localization: A Numerical Study". In: *Proc. International Conference on Robotics and Automation*. Philadelphia, USA, May 2022.

[6] Guoquan Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. "A First-Estimates Jacobian EKF for Improving SLAM Consistency". In: *Proc. of the 11th International Symposium on Experimental Robotics*. Athens, Greece, 2008.

[7] Guoquan Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. "Observability-based Rules for Designing Consistent EKF SLAM Estimators". In: *International Journal of Robotics Research* 29.5 (Apr. 2010), pp. 502–528. DOI: 10.1177/0278364909353640.

[8] Chuchu Chen, Yulin Yang, Patrick Geneva, and Guoquan Huang. "FEJ2: A Consistent Visual-Inertial State Estimator Design". In: *Proc. International Conference on Robotics and Automation*. Philadelphia, USA, May 2022.

[9] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. "The TUM VI benchmark for evaluating visual-inertial odometry". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1680–1687.