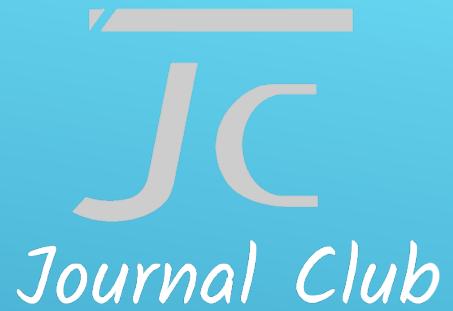




iMorpheus.ai

Weekly Journal Club



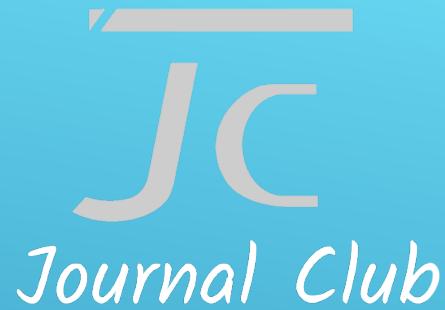


Journal club介绍与自动驾驶中定位方案相关的论文，主要关注的方向有：
SLAM算法、点云数据的处理和压缩、特征地图、传感器数据处理和融合、**GNSS**信号处理等。我们一直关注领域前沿技术，选取得到广泛认可的、或者是在我们的实际使用中结果比较好的论文，与大家分享，共同学习成长。

每周五 北京时间12点
<http://imorpheus.ai/journalclub>



扫码加入无人驾驶技术群



在SLAM计算中，会遇到让感知器失效的特定环境。黑夜、光线的变化、无结构特征等场景，会导致视觉相机的失效；单一几何特征的场景，会导致激光雷达的失效；这些场景都会导致SLAM出现严重偏差，即使是组合感知器的SLAM。

本周和大家一起学习讨论Zhang Ji 2017年提出一种加强组合SLAM方法。该方法硬件上集合了IMU, Camera, Lidar三种感知器，算法上运用了V-LOAM，以及在2016年提出的简并环境下的探测算法，通过提出的组合序贯模型来进行更加鲁棒的SLAM。

关键词：IMU Prediction, V-LOAM, Degradation

每周五 北京时间12点

<http://imorpheus.ai/journalclub>



扫码加入无人驾驶技术群

ENABLING AGGRESSIVE MOTION ESTIMATION AT LOW-DRIFT AND ACCURATE MAPPING IN REAL-TIME

TC. LIU, iMorpheus.ai

2017-12-29



KITTI-Visual Odometry / SLAM Evaluation 2012

Additional information used by the methods

- ☒ Stereo: Method uses left and right (stereo) images
- ☒ Laser Points: Method uses point clouds from Velodyne laser scanner
- ☐ Loop Closure Detection: This method is a SLAM method that detects loop closures
- ☒ Additional training data: Use of additional data sources for training (see details)

	Method	Setting	Code	<u>Translation</u>	Rotation	Runtime	Environment	Compare
1	V-LOAM	☒		0.63 %	0.0014 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
J. Zhang and S. Singh: Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast . IEEE International Conference on Robotics and Automation(ICRA) 2015.								
2	LOAM	☒		0.64 %	0.0014 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
J. Zhang and S. Singh: LOAM: Lidar Odometry and Mapping in Real-time . Robotics: Science and Systems Conference (RSS) 2014.								

17	DEMO	☒		1.14 %	0.0049 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
J. Zhang, M. Kaess and S. Singh: Real-time Depth Enhanced Monocular Odometry . IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2014.								



Zhang Ji's Papers

标题	引用次数	年份
Aerial and Ground-based Collaborative Mapping: An Experimental Study. J Zhang, S Singh The 11th International Conference on Field and Service Robotics	2017	
Enabling Aggressive Motion Estimation at Low-drift and Accurate Mapping in Real-time J Zhang, S Singh IEEE International Conference on Robotics and Automation (ICRA)	4	2017
Low-drift and Real-time Lidar Odometry and Mapping J Zhang, S Singh Autonomous Robots 41 (2), 401-416	16	2017
A Real-time Method for Depth Enhanced Visual Odometry J Zhang, M Kaess, S Singh Autonomous Robots 41 (1), 31-43	5	2017
On Degeneracy of Optimization-based State Estimation Problems J Zhang, M Kaess, S Singh IEEE International Conference on Robotics and Automation (ICRA)	4	2016
anyFish 2.0: An open-source software platform to generate and share animated fish models to study behavior S Ingley, M Asl, C Wu, R Cui, M Gadelhak, W Li, J Zhang, J Simpson, ... SoftwareX 3, 13–21	8	2015
Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast J Zhang, S Singh IEEE International Conference on Robotics and Automation (ICRA)	80	2015
Visual-Inertial Combined Odometry System for Aerial Vehicles J Zhang, S Singh Journal of Field Robotics	5	2015
Robot Farmers: Autonomous Orchard Vehicles Help Tree Fruit Production M Bergerman, S Maeta, J Zhang, G Freitas, B Hamner, S Singh, K George IEEE Robotics and Automation Magazine 22 (1), 54-63	14	2015
Real-time Depth Enhanced Monocular Odometry J Zhang, M Kaess, S Singh IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)	48	2014
Mapping orchards for autonomous navigation J Zhang, S Maeta, M Bergerman, S Singh Proc. American Society of Agricultural and Biological Engineers Annu. Int ...	9	2014
LOAM: Lidar Odometry and Mapping in Real-time J Zhang, S Singh Robotics: Science and Systems Conference (RSS 2014)	149	2014



Zhang Ji's Papers

2014

- 1) Real-time Depth Enhanced Monocular Odometry. -> DEMO
- 2) LOAM: Lidar Odometry and Mapping in Real-time. -> LOAM

2015

- 1) Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast. -> V-LOAM
- 2) Visual–Inertial Combined Odometry System for Aerial Vehicles



iMorpheus.ai

Zhang Ji's Papers

2016

- 1) On Degeneracy of Optimization-based State Estimation Problems.

2017

- 1) Low-drift and real-time lidar odometry and mapping.
- 2) A real-time method for depth enhanced visual odometry.
- 3) **Enabling Aggressive Motion Estimation at Low-drift and Accurate Mapping in Real-time.**



iMorpheus.ai

Zhang Ji's Papers

Review:

<http://www.imorpheus.ai/journalClub>

<https://github.com/iMorpheusAI/journalClub>

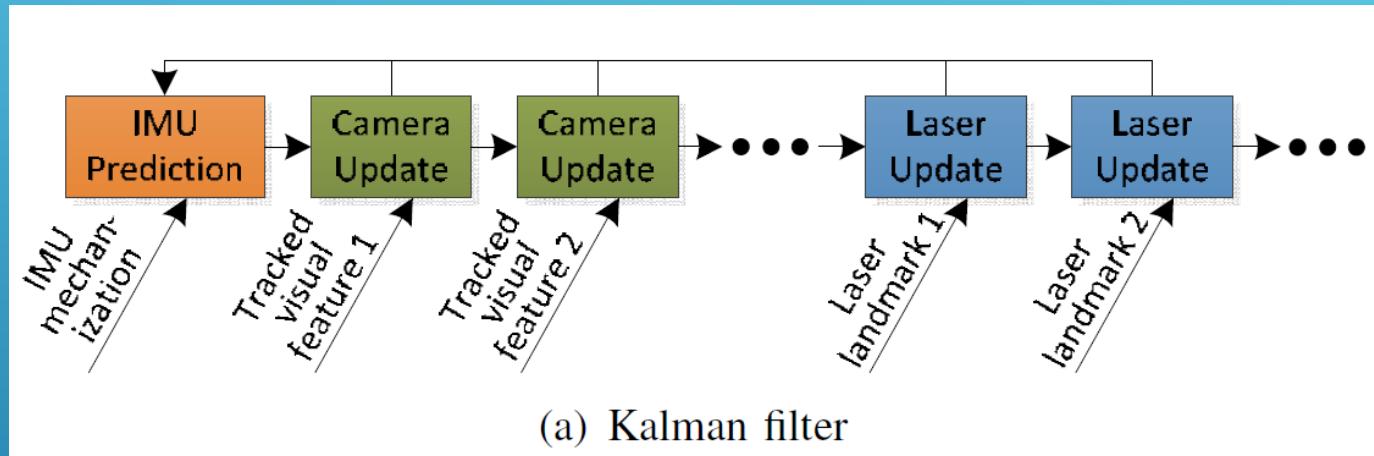
Simon -> V-LOAM

YuLu -> An optimization-based method in state estimation



iMorpheus.ai

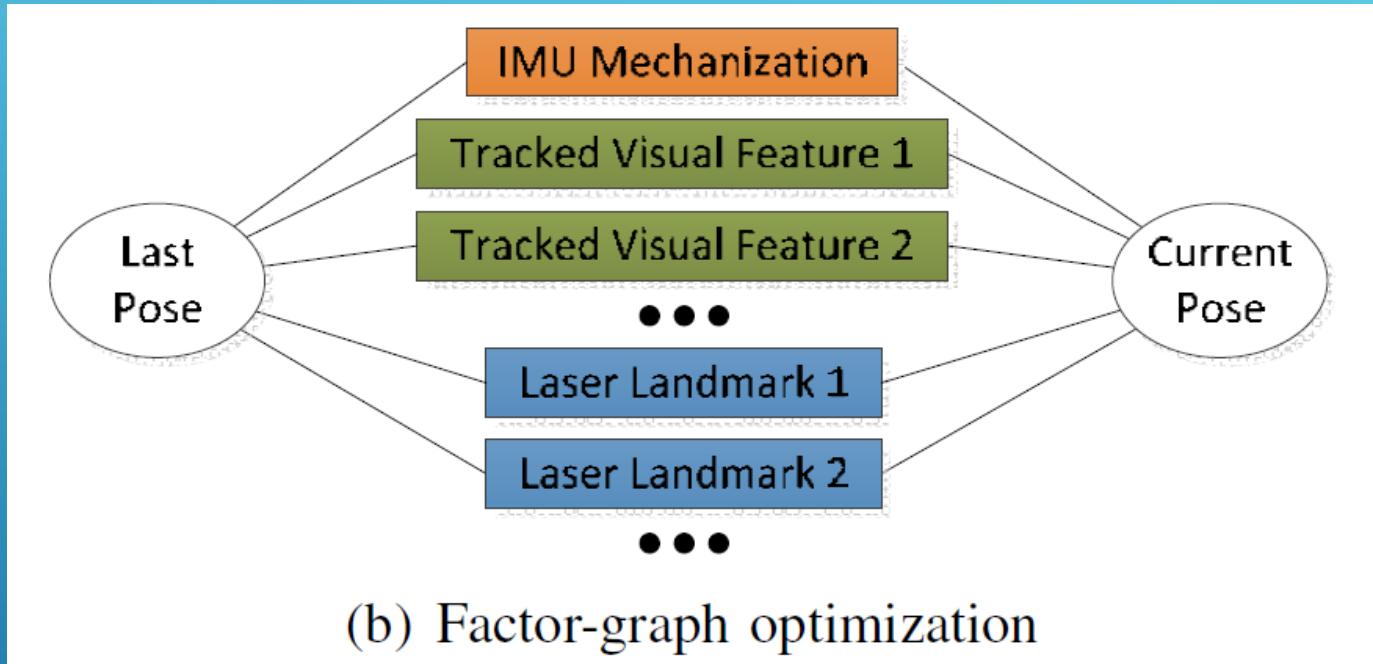
Model-1



A Kalman filter based method typically processes individual visual features and laser landmarks in separate steps.



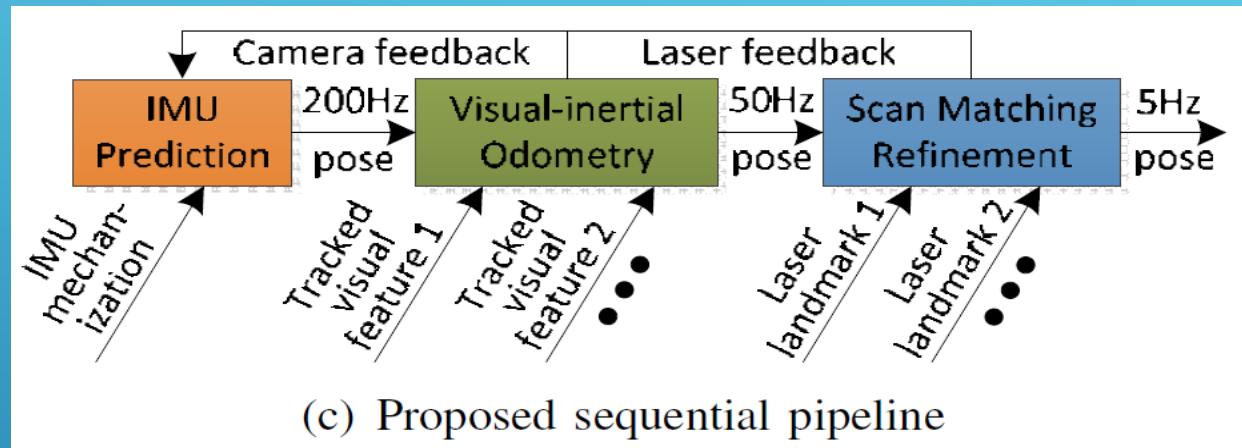
Model-2



A factor-graph optimization based method combines all sensor data into a full-blown optimization problem.



Model-3



Our system recovers motion through multilayer processing in a coarse-to-fine manner.

Starting with motion prediction from an IMU, a visual inertial coupled method estimates motion and registers laser points locally.

A scan matching method refines the estimated motion and registers point clouds.



Advantages

Calibration & Computation

High-frequency modules are specialized to handle aggressive motion, while low frequency modules cancel drift from the previous modules.

Modules in the front take less computation and execute at high frequencies.

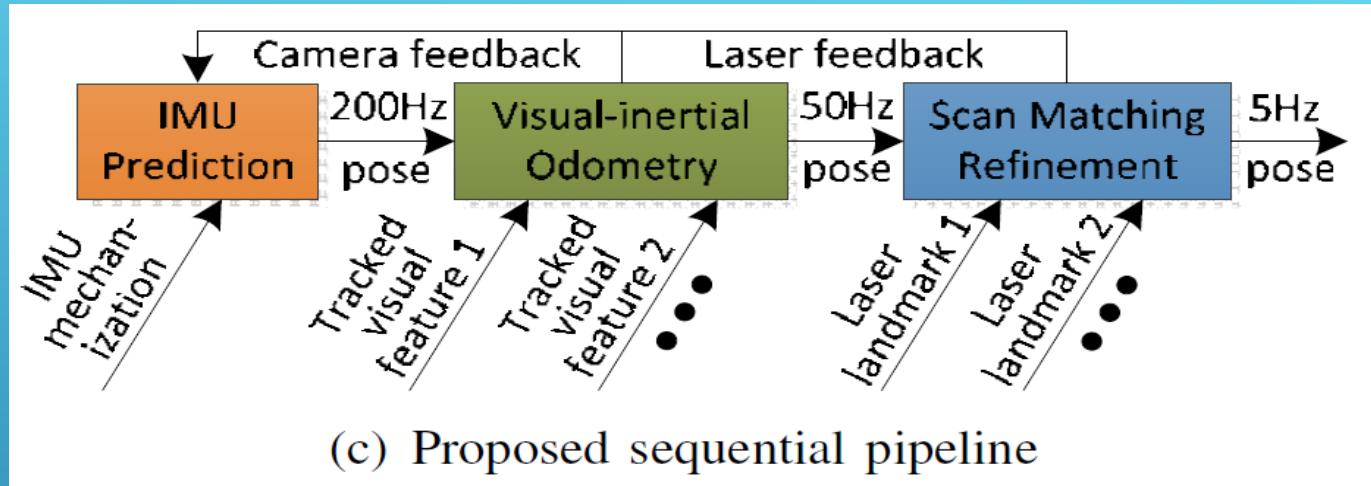
Handle sensor degradation

If the camera is non-functional, e.g.
due to darkness, dramatic lighting changes, or texture-less environments

If the laser is non-functional, e.g.
due to structure-less environments

Even both are non-functional





KeyPoint:

IMU:

Prediction+ Feedback

Camera+Lidar:

Visual-inertial Odometry+ Lidar Odometry+ Refinement

Robustness:

Degradation method

IMU Prediction Subsystem

The IMU measurement terms:

$$\hat{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) + \mathbf{b}_{\boldsymbol{\omega}}(t) + \mathbf{n}_{\boldsymbol{\omega}}(t), \quad (1)$$

$$\hat{\mathbf{a}}(t) = \mathbf{a}(t) - {}_W^I \mathbf{R}(t) \mathbf{g} + \mathbf{b}_{\mathbf{a}}(t) + \mathbf{n}_{\mathbf{a}}(t), \quad (2)$$

Let $\boldsymbol{\omega}(t)$ and $\mathbf{a}(t)$ be two 3×1 vectors indicating the angular rates and accelerations of the IMU frame $\{I\}$. Let $\mathbf{b}_{\boldsymbol{\omega}}(t)$ and $\mathbf{b}_{\mathbf{a}}(t)$ be the corresponding biases, and $\mathbf{n}_{\boldsymbol{\omega}}(t)$ and $\mathbf{n}_{\mathbf{a}}(t)$ be the noises. Additionally, let \mathbf{g} be the constant gravity vector in the world frame $\{W\}$

${}_W^I \mathbf{R}(t)$ is the rotation matrix from $\{W\}$ to $\{I\}$.



IMU Prediction Subsystem

Feedback In IMU is simple:

The IMU bias correction can be made by feedback from either the camera or the laser. By comparing the estimated motion with IMU integration, we can calculate $b_\omega(t)$ and $b_a(t)$. To reduce high-frequency noises, a sliding window is employed keeping a certain number of biases. The averaged terms are used.



Visual-Inertial Odometry Subsystem

The Workflow :

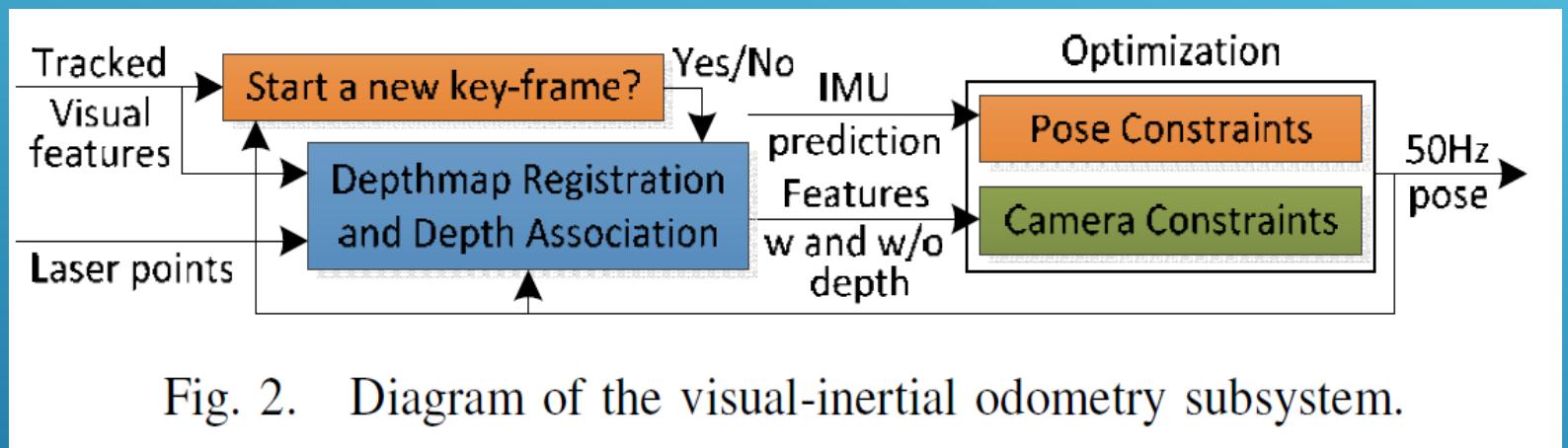


Fig. 2. Diagram of the visual-inertial odometry subsystem.



Visual-Inertial Odometry Subsystem

The Motion Function:

$$\mathbf{X}_c = \mathbf{R}_l^c \mathbf{X}_l + \mathbf{t}_l^c. \quad (3)$$

Let us define \mathbf{T}_a^b as a 4×4 matrix representing the motion transform between frames a and b , \mathbf{T}_a^b corresponds to a set of \mathbf{R}_a^b and \mathbf{t}_a^b .

To formulate the IMU pose constraints, we take the solved motion transform between frames l and $c-1$, namely \mathbf{T}_l^{c-1} . From IMU mechanization, we obtain a predicted transform between the last two frames $c-1$ and c , denoted as $\hat{\mathbf{T}}_{c-1}^c$. The predicted transform at frame c is $\hat{\mathbf{T}}_l^c = \hat{\mathbf{T}}_{c-1}^c \mathbf{T}_l^{c-1}$.



Visual-Inertial Odometry Subsystem

The Refine Optimization:

$$\Sigma_l^c [(\hat{\theta}_l^c - \theta_l^c)^T, (\hat{t}_l^c(\theta_l^c) - t_l^c)^T]^T = \boldsymbol{0}, \quad (7)$$

where Σ_l^c is a relative covariance matrix scaling the pose constraint appropriately w.r.t. the camera constraints.



Scan Matching Subsystem

The Workflow :

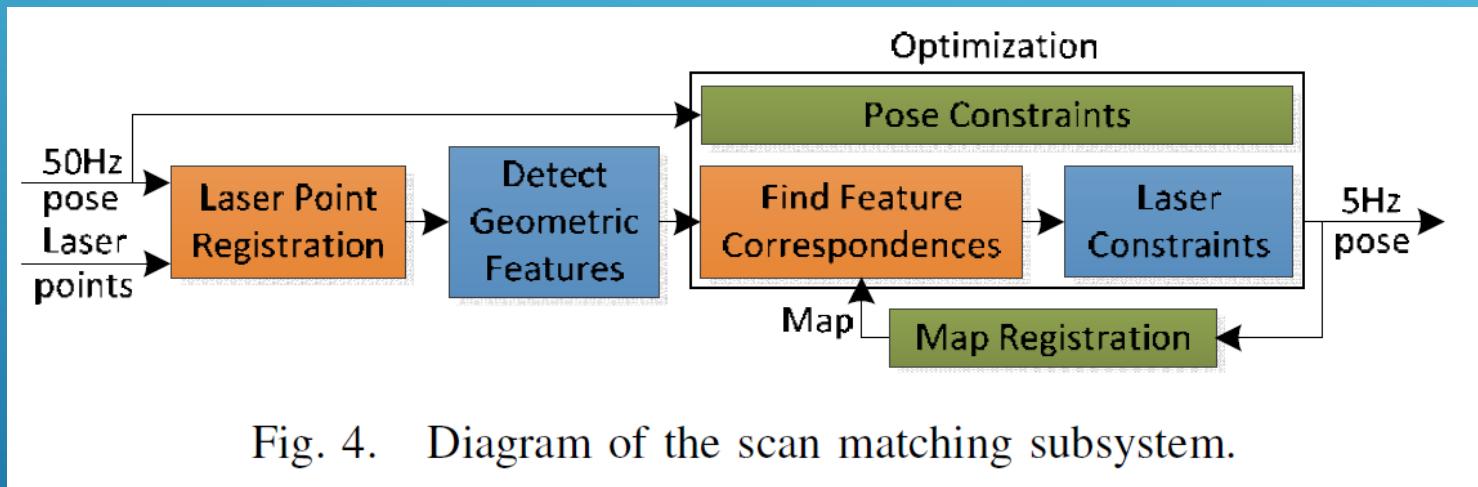


Fig. 4. Diagram of the scan matching subsystem.



Scan Matching Subsystem

The Optimization Problem:

$$d = f(X_m, \theta_m, t_m), \quad (8)$$

The Refinement:

$$\Sigma_m [(\hat{\theta}_m - \theta_m)^T, (\hat{t}_m - t_m)^T]^T = \boldsymbol{0}. \quad (9)$$



System Robustness

Hardware robustness

Camera degradation | | Laser degradation

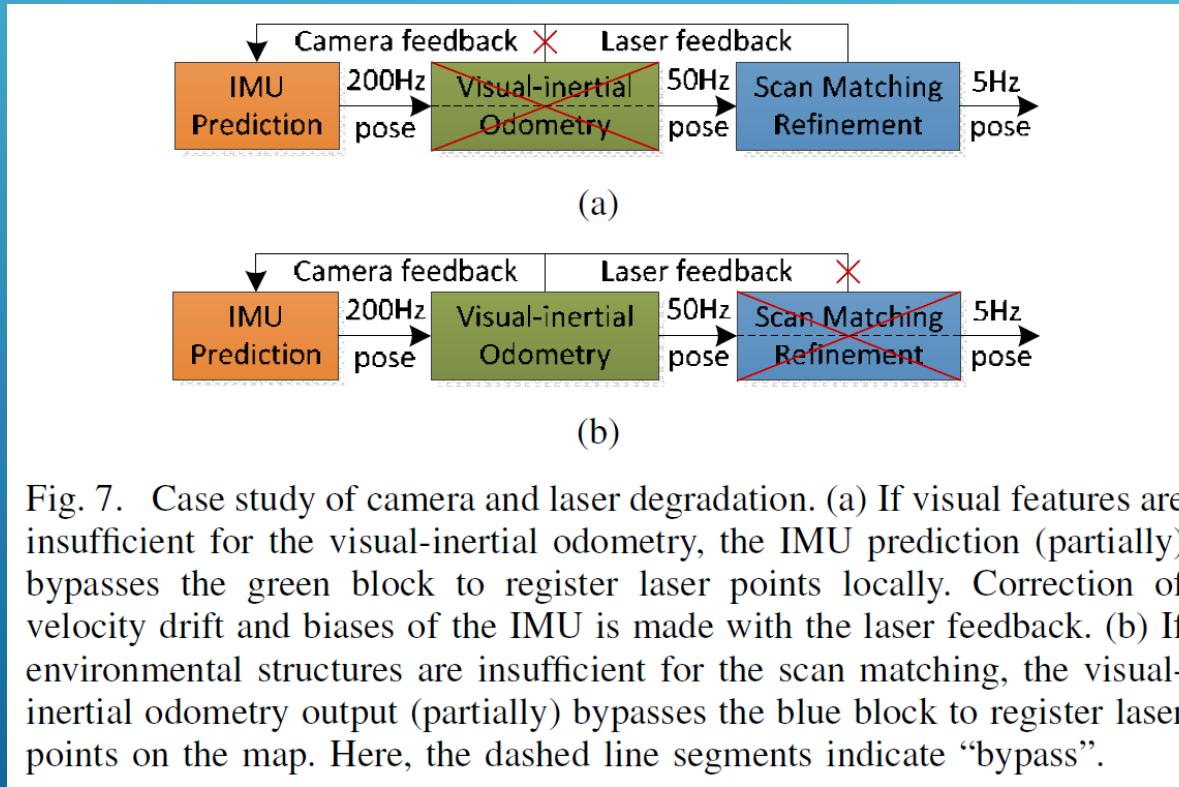


Fig. 7. Case study of camera and laser degradation. (a) If visual features are insufficient for the visual-inertial odometry, the IMU prediction (partially) bypasses the green block to register laser points locally. Correction of velocity drift and biases of the IMU is made with the laser feedback. (b) If environmental structures are insufficient for the scan matching, the visual-inertial odometry output (partially) bypasses the blue block to register laser points on the map. Here, the dashed line segments indicate “bypass”.



System Robustness

Hardware robustness

Camera degradation && Laser degradation

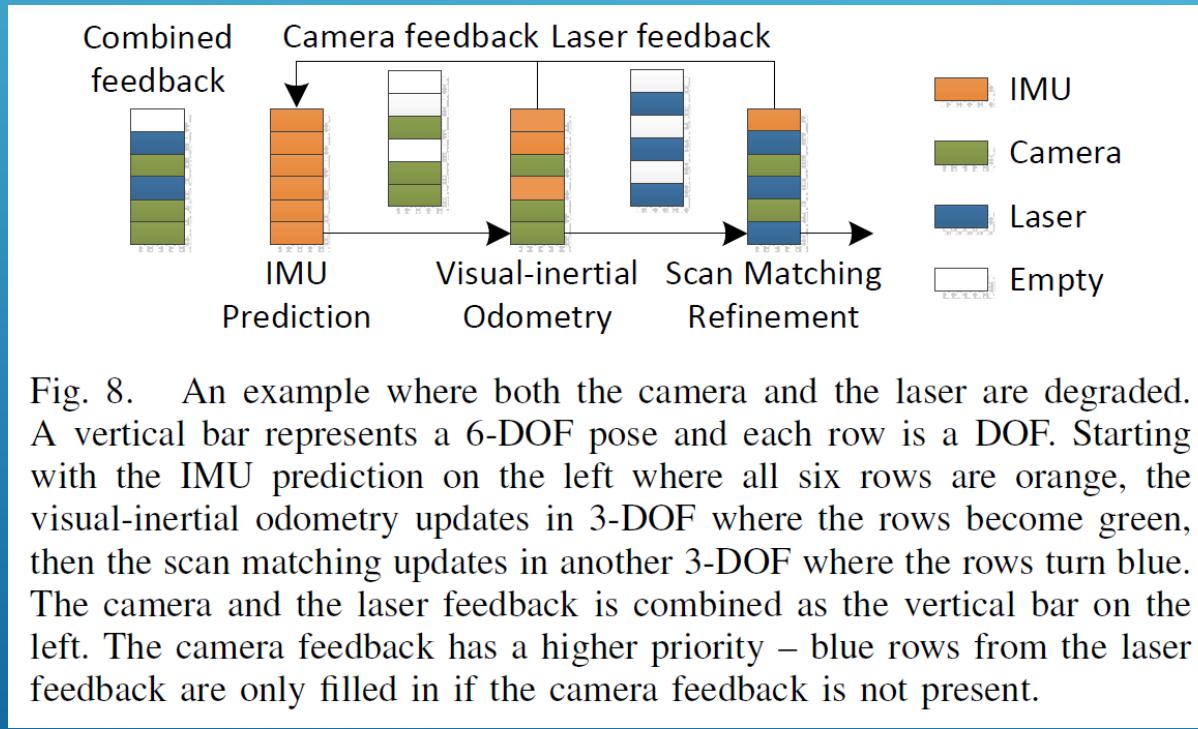


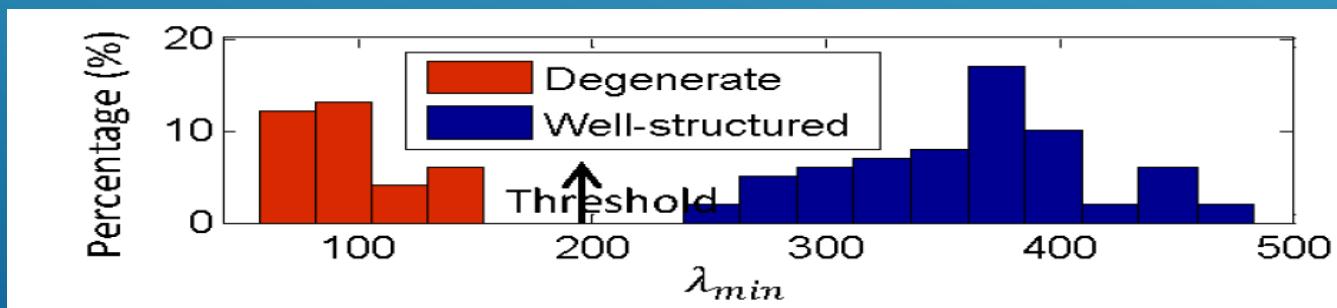
Fig. 8. An example where both the camera and the laser are degraded. A vertical bar represents a 6-DOF pose and each row is a DOF. Starting with the IMU prediction on the left where all six rows are orange, the visual-inertial odometry updates in 3-DOF where the rows become green, then the scan matching updates in another 3-DOF where the rows turn blue. The camera and the laser feedback is combined as the vertical bar on the left. The camera feedback has a higher priority – blue rows from the laser feedback are only filled in if the camera feedback is not present.



System Robustness

Software robustness Detection

- In linearized problem $\arg \min_x \|\mathbf{Ax} - \mathbf{b}\|^2$, we can assume matrix A is non singular, thus $A^T A$ is positive definite
- Eigenvalues of $A^T A$ are listed, in increasing order, as $\lambda_1, \dots, \lambda_n$ and the corresponding eigenvectors are v_1, \dots, v_n
- Find certain threshold to determine which dimensions are degenerate



System Robustness

Software robustness Correction

$$\begin{aligned}\mathbf{V}_p &= [\mathbf{v}_1, \dots, \mathbf{v}_m, 0, \dots, 0]^T, \\ \mathbf{V}_u &= [0, \dots, 0, \mathbf{v}_{m+1}, \dots, \mathbf{v}_n]^T, \\ \mathbf{V}_f &= [\mathbf{v}_1, \dots, \mathbf{v}_m, \mathbf{v}_{m+1}, \dots, \mathbf{v}_n]^T\end{aligned}$$

- x_p is the best guess for true state and x_u is the unaltered solution
- x_p is given by some a priori model, such as constant velocity model
 - The system solves for motion in a coarse-to-fine order, starting with the IMU prediction.
- The final solution is $\mathbf{x}_f = \mathbf{x}'_p + \mathbf{x}'_u$, where $\mathbf{x}'_p = \mathbf{V}_f^{-1} \mathbf{V}_p \mathbf{x}_p$ and $\mathbf{x}'_u = \mathbf{V}_f^{-1} \mathbf{V}_u \mathbf{x}_u$



Experiments

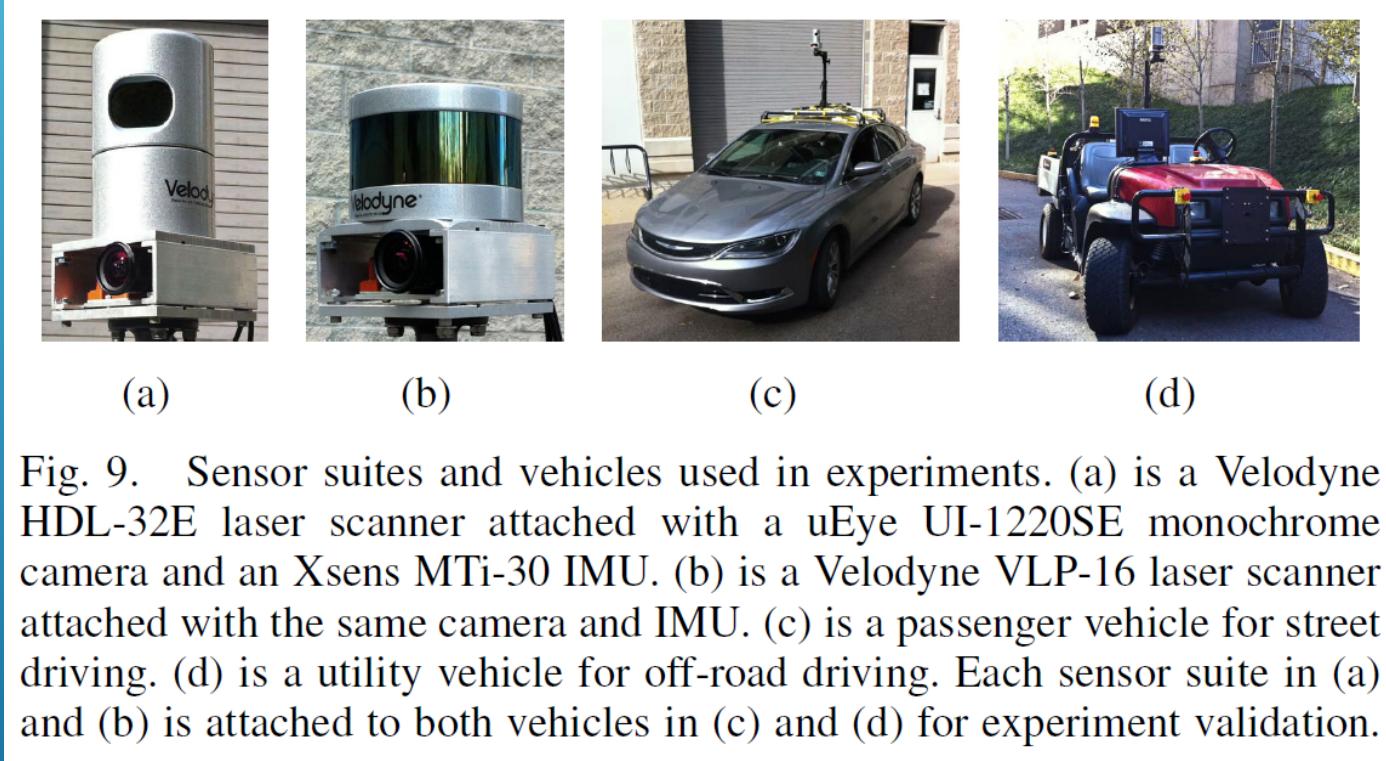


Fig. 9. Sensor suites and vehicles used in experiments. (a) is a Velodyne HDL-32E laser scanner attached with a uEye UI-1220SE monochrome camera and an Xsens MTi-30 IMU. (b) is a Velodyne VLP-16 laser scanner attached with the same camera and IMU. (c) is a passenger vehicle for street driving. (d) is a utility vehicle for off-road driving. Each sensor suite in (a) and (b) is attached to both vehicles in (c) and (d) for experiment validation.



Experiments - Accuracy Test

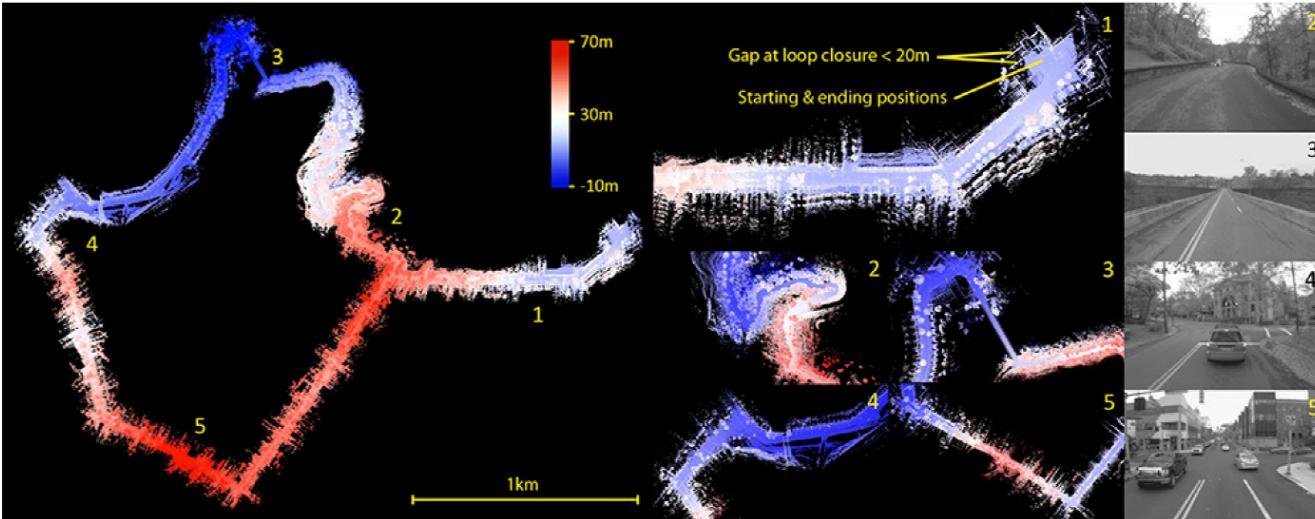


Fig. 11. Accuracy test. The sensor suite in Fig. 9(a) is mounted on the vehicle in Fig. 9(c) for 9.3km of street driving. The path goes through vegetated environments, bridges, hilly terrains, and roads with heavy traffic. The elevation changes over 70m. Except waiting for traffic lights, the vehicle is driven at 9-18m/s. On the left, we show the complete map color coded by elevation. On the right, we show a few close views with corresponding locations labeled with numbers 1-5 on the map. In close view 1, we present the starting and the ending positions. Because of drift, a building is registered into two, one during the vehicle leaves from the start and the other during the vehicle returns at the end. We manually measure the gap to be $< 20m$, resulting in a relative position error at the end to be $< 0.22\%$ of the distance traveled. Close views 2-5 show more details with images logged by the camera.



iMorpheus.ai

Experiments - Accuracy Test

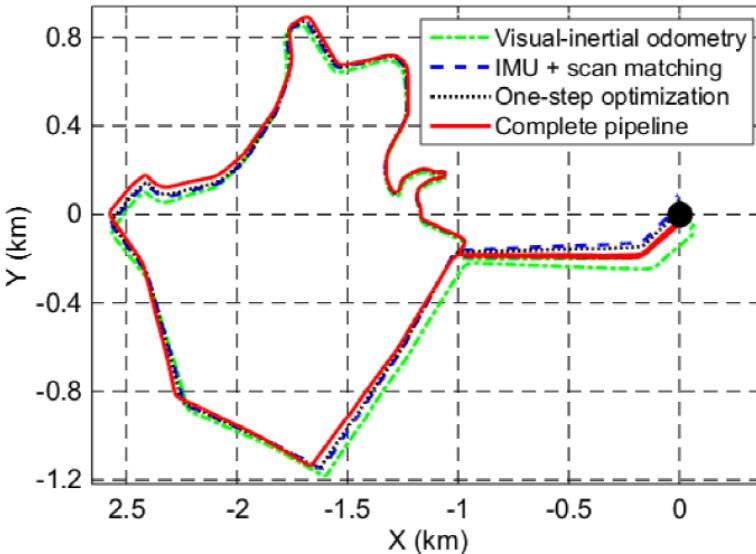


Fig. 12. Estimated trajectories in accuracy test. The trajectories start with the black dot. We compare four system configurations in the test. The green dash-dot curve is from the visual-inertial odometry module (using the left two modules in Fig. 1(c)). The blue dash curve is from the scan matching module with the IMU prediction directly taken as input (leftmost and rightmost modules in Fig. 1(c)). The black dot curve has the system reconfigured to solve one large optimization problem incorporating all constraints, as in Fig. 1(b). The red solid curve is from the proposed data processing pipeline.



Experiments - Robustness Test

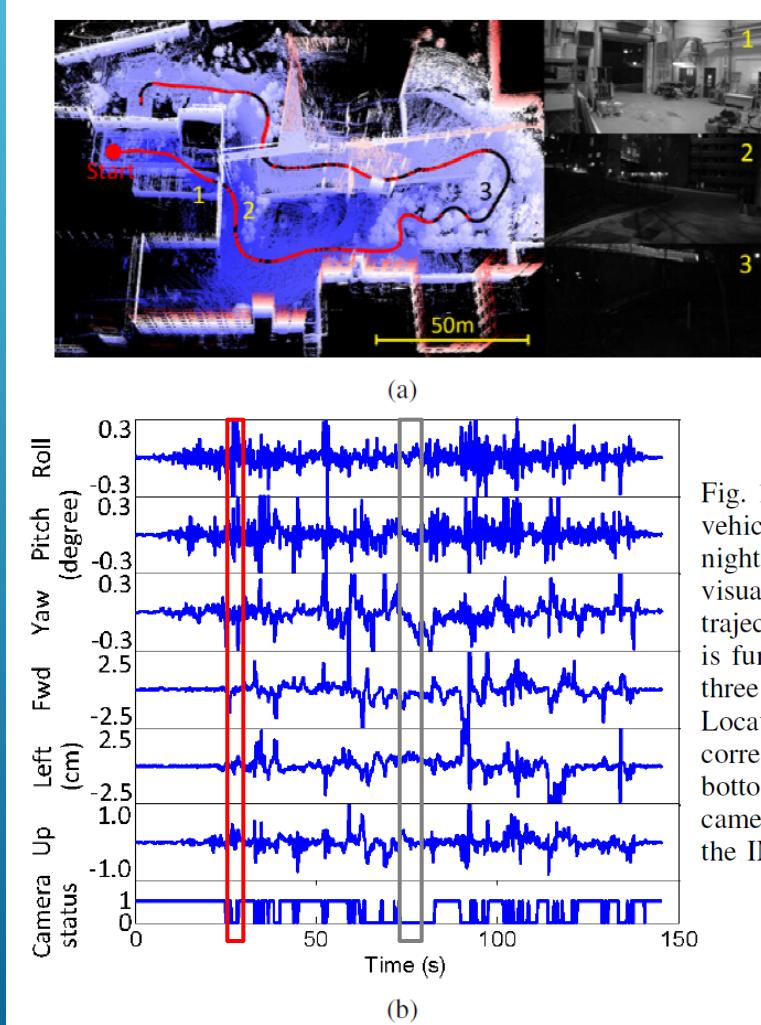


Fig. 13. Robustness test 1. The sensor suite in Fig. 9(b) is attached to the vehicle in Fig. 9(d) driven from indoor to outdoor. The test is conducted at night. Frequently, the camera cannot capture enough visual features and the visual-inertial odometry module is bypassed. In (a), we show the estimated trajectory overlayed on the map built. The red segments indicate vision is functional and the black segments indicate degradation. Also, we show three images logged by the camera from locations 1-3 labeled on the map. Location 1 is indoor and locations 2-3 are outdoor. In (b), we show pose corrections applied by the scan matching to refine motion estimates. On the bottom row, the camera status being one indicates functioning. When the camera status is zero, corrections on the top six rows become larger because the IMU prediction produces more drift than the visual-inertial odometry.



Experiments - Robustness Test

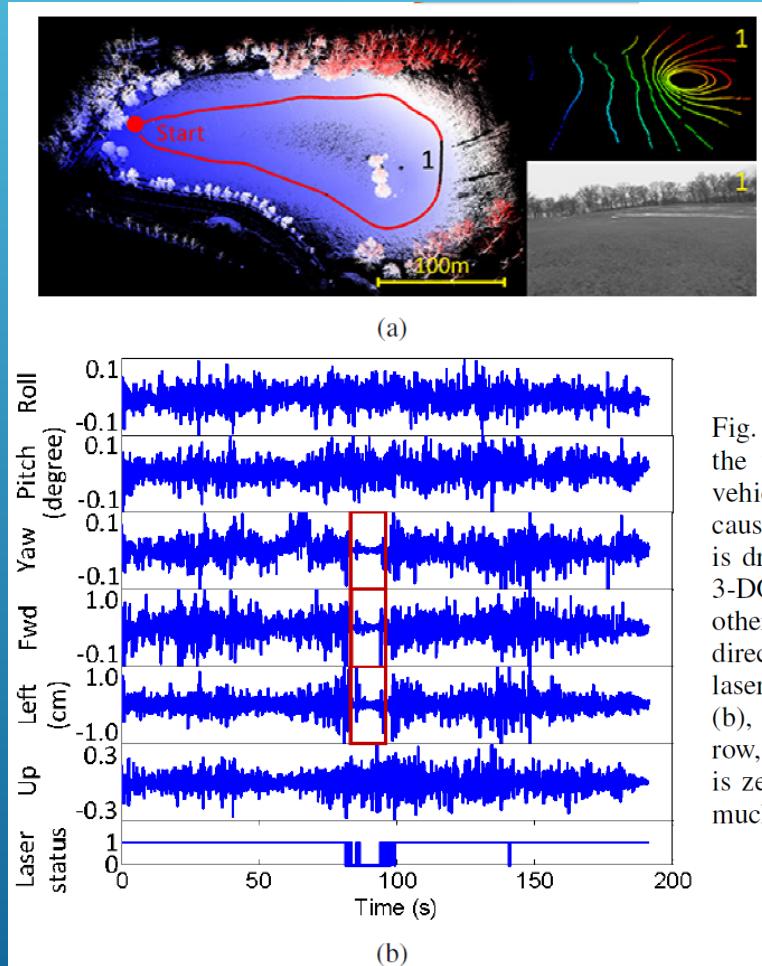


Fig. 14. Robustness test 2. The sensor suite in Fig. 9(b) is attached to the vehicle in Fig. 9(d) driven in an off-road terrain. In (a), when the vehicle reaches the rightmost side of the path, only the flat ground is seen, causing the scan matching to partially degrade. The corresponding trajectory is drawn in black. Here, we determine the scan matching is able to refine 3-DOF out of the 6-DOF motion, which are roll, pitch, and elevation. The other 3-DOF are unsolvable due to the planar scene, where the pose is directly taken from the visual-inertial odometry. In addition, we show a laser scan and an image logged from location 1 labeled on the map. In (b), we show pose corrections applied by the scan matching. On the last row, the laser status being one indicates functioning. When the laser status is zero, pose corrections in degraded directions (in the red boxes) become much smaller.



Final Thoughts

□ A robust pipeline should couple a 3D laser, a camera, and an IMU.

IMU is always reliable functioning as the backbone in the system.

□ Sequential model is deserve to dig for:

High frequency -> Low frequency

Low computation -> High computation

Coarse estimation -> Accurate estimation

Refinement && Feedback

□ Accumulative errors are still alive.

GNSS system is independent of x, y, z, t, which should be a good choice.

