OpenVINS: A Research Platform for Visual-Inertial Estimation

Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang

Robot Perception and Navigation Group (RPNG)

University of Delaware, USA

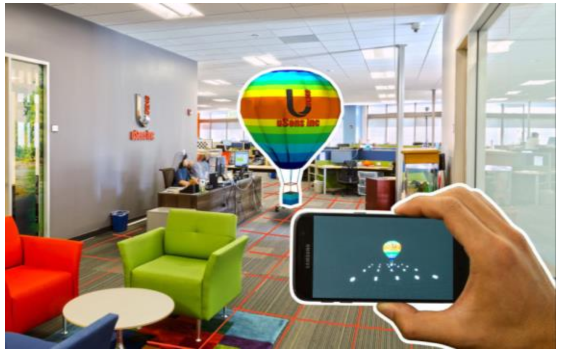
Motivation

- State estimation is crucial for many applications
- Algorithm ***complexity*** and ***accuracy*** are key to providing useable results at the edge

Extraterrestrial Robots

Autonomous Driving

Wearables and Health Tracking



AR / VR Experiences

Micro Aerial Vehicles

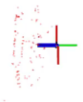
Human Pose Tracking

Warehouse Robotics

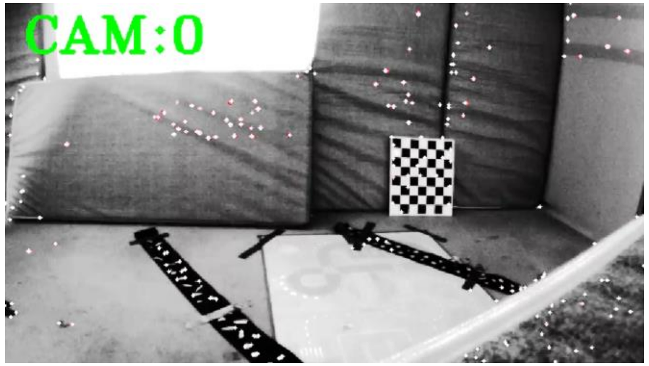
Nano Aerial Vehicles

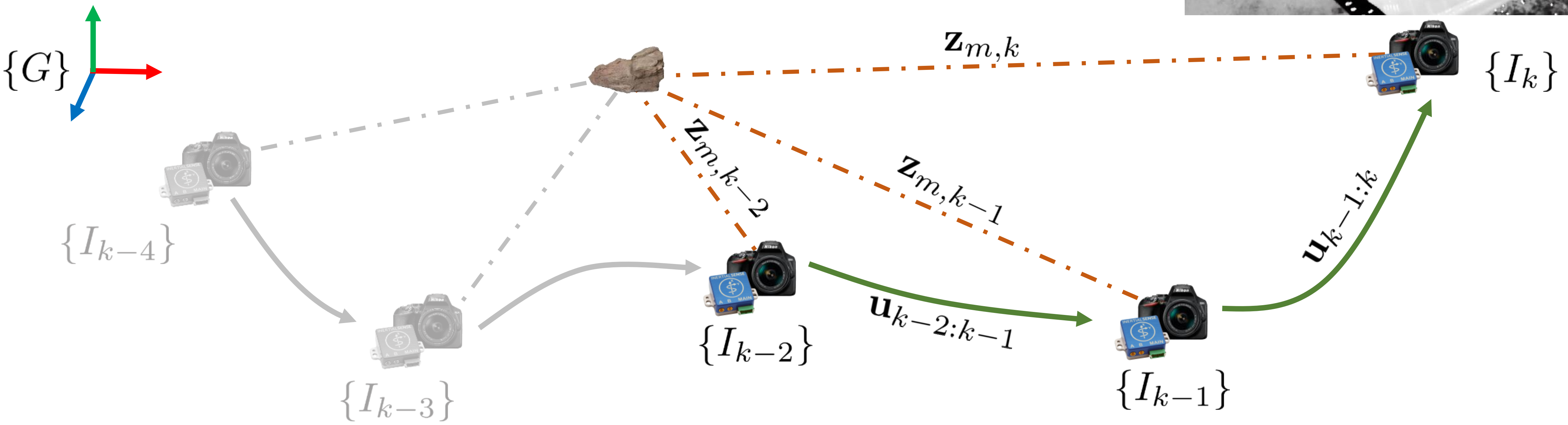
Visual-Inertial 3D Motion Tracking

- Visual-inertial sensor can provide ***low-cost*** and ***light-weight*** 3D localization
- Need to have state estimation algorithms which can fuse this information
- ***Visual-inertial navigation systems (VINS)*** can provide the solution



Problem: Visual-Inertial Estimation





- Goal: To estimate poses $\mathbf{x}_C = \begin{bmatrix} {}^{I_{k-1}}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_{k-1}}^\top & \cdots & {}^{I_{k-2}}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_{k-2}}^\top \end{bmatrix}^\top$ and inertial state $\mathbf{x}_I = \begin{bmatrix} {}^{I_k}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_k}^\top & {}^{G}\mathbf{v}_{I_k}^\top & \mathbf{b}_{\omega_k}^\top & \mathbf{b}_{a_k}^\top \end{bmatrix}^\top$ given the inertial readings $\mathbf{u}_{k-2:k}$ and bearings $\mathbf{z}_{m,k-2:k}$ of environmental features

Visual-Inertial Research: Embracing Open Source

Table 1: Open sourced visual-inertial estimation systems.

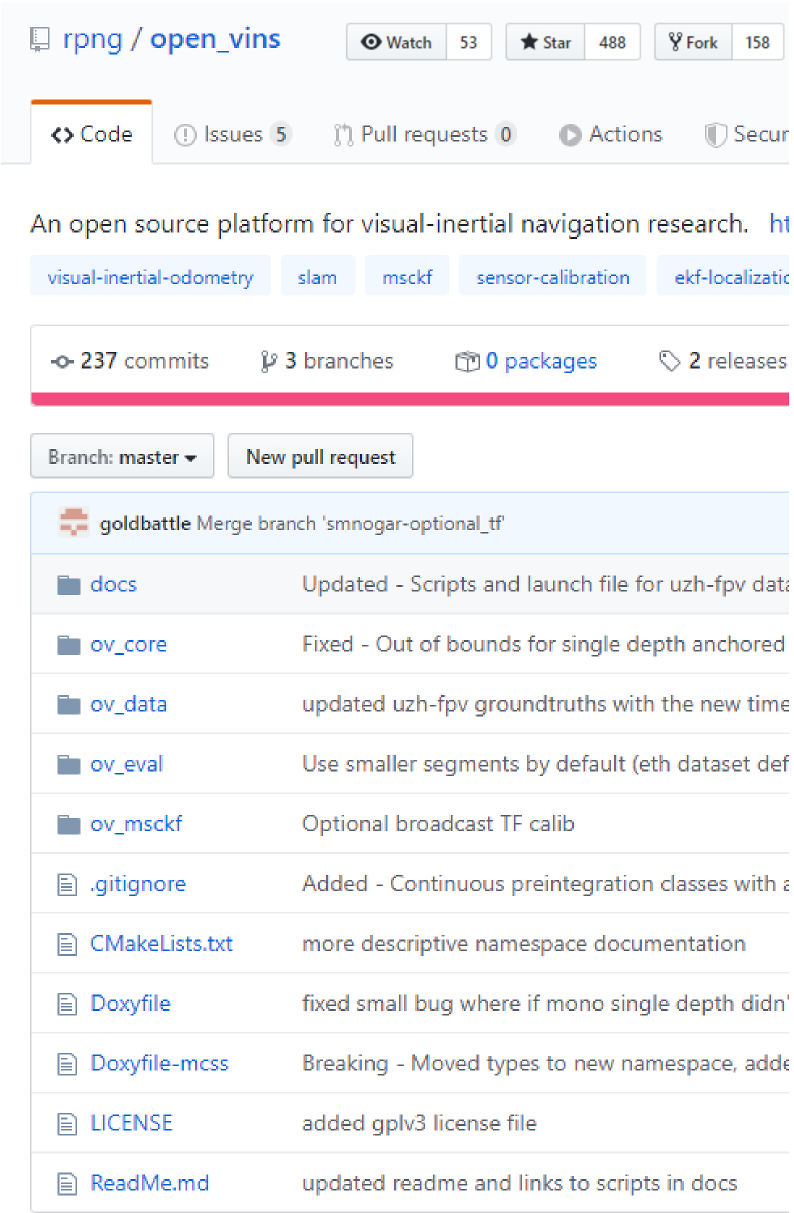
System	Mono?	Stereo?	EKF?	Geom?	Calib. Spacial?	Calib. Intrinsics?	Calib. Time?
S-MSCKF	✗	✓	✓	✓	✓	✗	✗
R-VIO	✓	✗	✓	✓	✗	✗	✗
Rovioli	✓	✗	✓	✗	✓	✗	✗
VINS-Fusion	✓	✓	✗	✓	✓	✗	✓
OKVIS	✓	✓	✗	✓	✓	✗	✗
Basalt	✗	✓	✗	✓	✗	✗	✗
ICE-BA	✗	✓	✗	✓	✗	✗	✗
OpenVINS	✓	✓	✓	✓	✓	✓	✓

- Wide range of systems available for visual-inertial research
- None provide a ***feature complete filter system*** with the accuracy of batch-based methods for use on resource constrained platforms

OpenVINS

- An open platform for VINS research (OpenVINS) which achieves state-of-the-art performance
- On manifold sliding window Kalman filter with modular ***type system*** for state management

https://github.com/rpng/open_vins
https://docs.openvins.com/



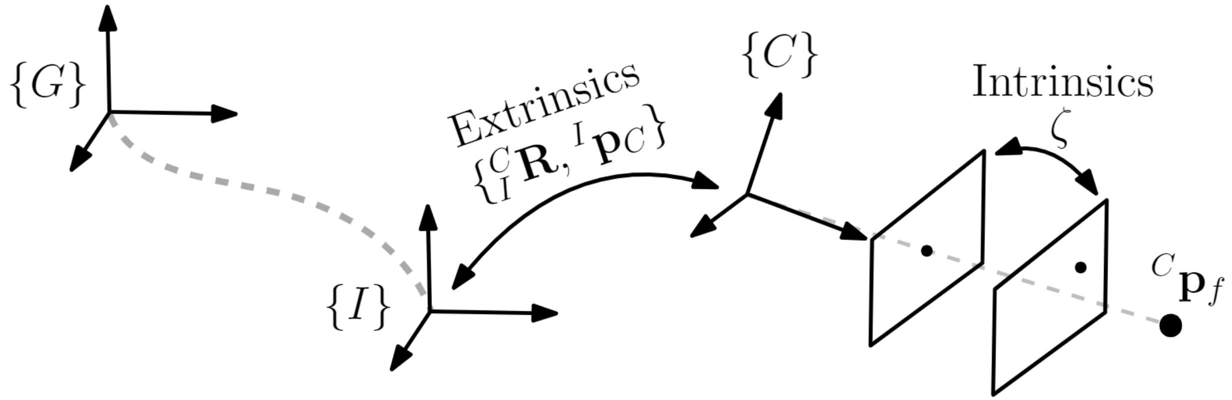
Key Feature – Type-Based State Management

- Each estimation variable is a type
- Indexes automatically managed during operations (augmentation, marginalization, cloning, etc.)
- ***Intuitive*** syntax for covariance access

```
covariance.block(
    imu->_id,imu->_id,
    imu->_size,imu->_size
);
```

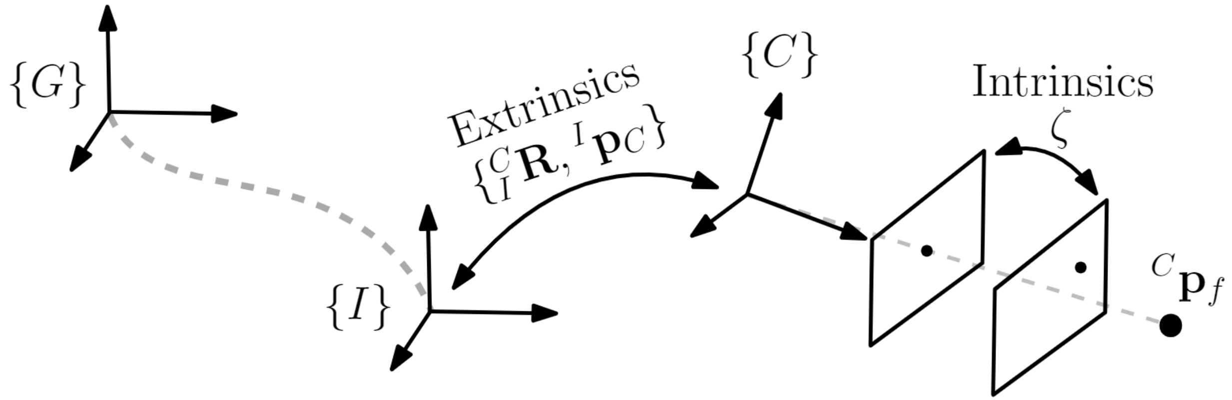
```
class Type {
protected:
  // Current best estimate
  Eigen::MatrixXd _value;
  // Index of error state in covariance
  int _id = -1;
  // Dimension of error state
  int _size = -1;
  // Vector correction, how to update
  void update(const Eigen::VectorXd dx);
};
```

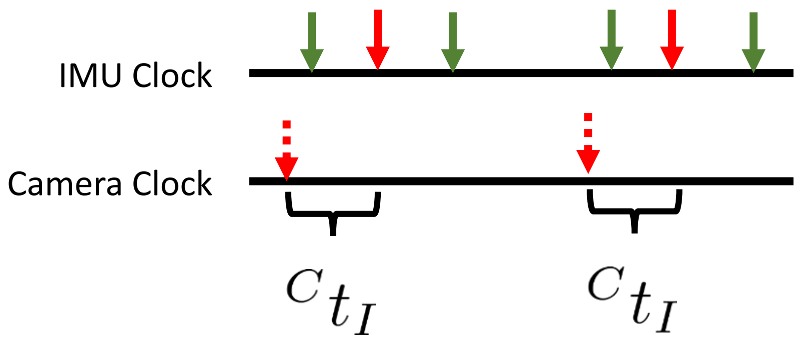
Key Feature – Online Calibration



- Calibration of camera intrinsics and extrinsics

Key Feature – Online Calibration



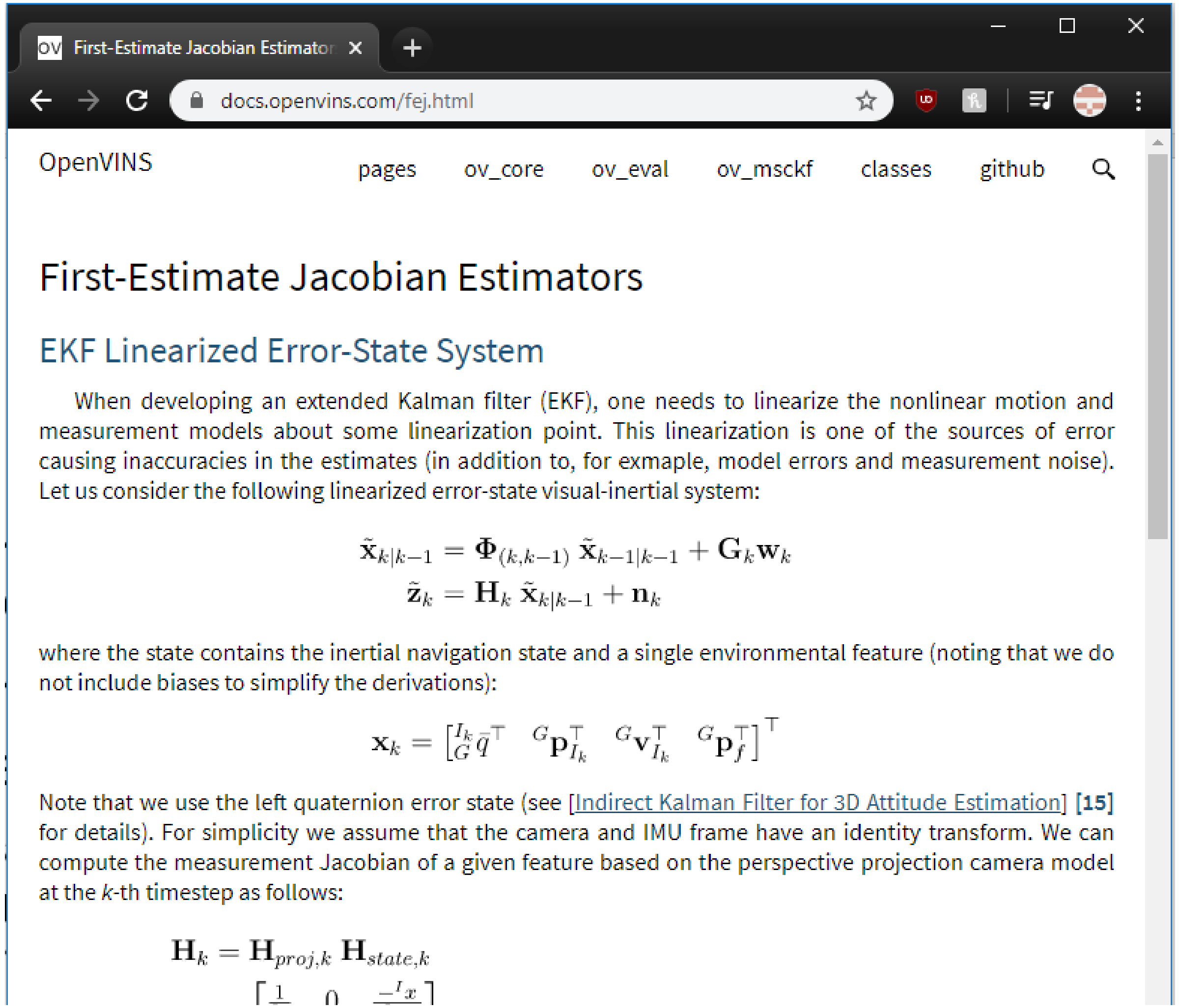


- Calibration of camera intrinsics and extrinsics
- Additionally, temporal offset between IMU and camera ${}^It = {}^Ct + {}^Ct_I$ is performed
- Crucial for ***practical deployments*** handling of poor initial values

Key Feature – First-Estimates Jacobians

- Temporal ***SLAM landmarks*** with First-Estimate Jacobians with six different representations

- Detailed ***documentation*** and derivations facilitating future research



First-Estimate Jacobian Estimator

docs.openvins.com/fej.html

OpenVINS pages ov_core ov_eval ov_msckf classes github

First-Estimate Jacobian Estimators

EKF Linearized Error-State System

When developing an extended Kalman filter (EKF), one needs to linearize the nonlinear motion and measurement models about some linearization point. This linearization is one of the sources of error causing inaccuracies in the estimates (in addition to, for exmaple, model errors and measurement noise). Let us consider the following linearized error-state visual-inertial system:

$$\begin{aligned}\tilde{\mathbf{x}}_{k|k-1} &= \mathbf{\Phi}_{(k,k-1)}\,\tilde{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_k\mathbf{w}_k \\ \tilde{\mathbf{z}}_k &= \mathbf{H}_k\,\tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}_k\end{aligned}$$

where the state contains the inertial navigation state and a single environmental feature (noting that we do not include biases to simplify the derivations):

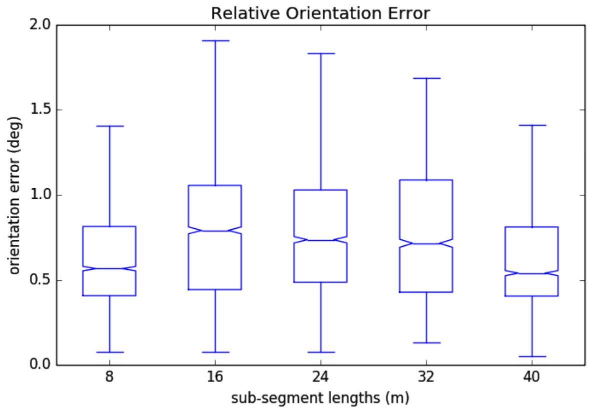
$$\mathbf{x}_k = \begin{bmatrix} {}^{I_k}_{G}\bar{q}^{\top} & {}^{G}\mathbf{p}_{I_k}^{\top} & {}^{G}\mathbf{v}_{I_k}^{\top} & {}^{G}\mathbf{p}_f^{\top}\end{bmatrix}^{\top}$$

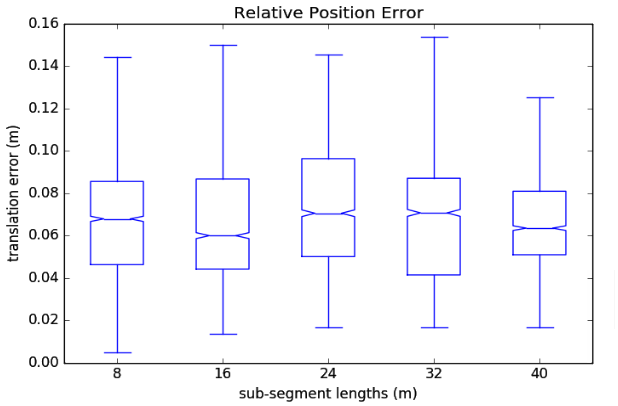
Note that we use the left quaternion error state (see [Indirect Kalman Filter for 3D Attitude Estimation] **[15]** for details). For simplicity we assume that the camera and IMU frame have an identity transform. We can compute the measurement Jacobian of a given feature based on the perspective projection camera model at the *k*-th timestep as follows:

$$\mathbf{H}_k = \mathbf{H}_{proj,k}\,\mathbf{H}_{state,k}$$

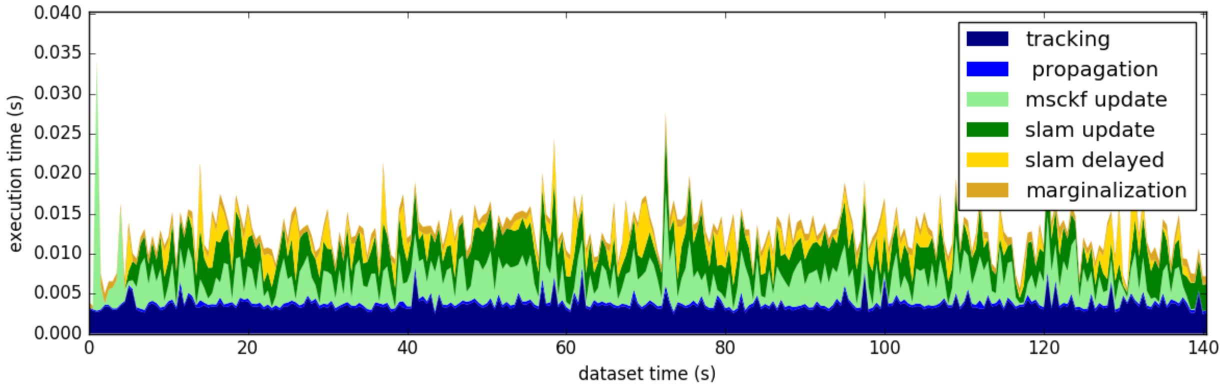
Key Feature – System Evaluation

- Many standard ***error metrics*** with plotting
  - ATE, RPE, NEES, RMSE
  - Monte-Carlo run support



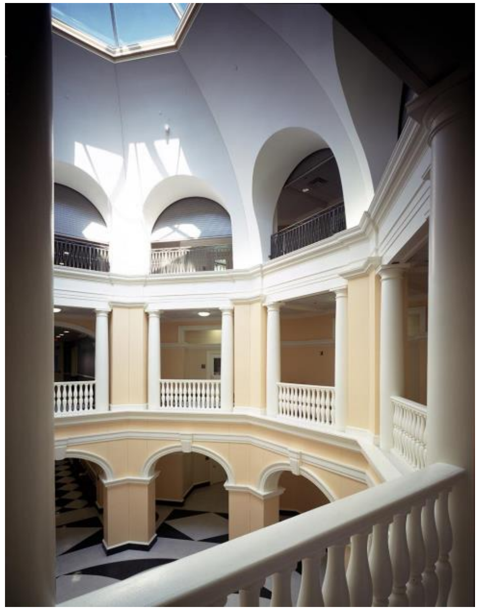


- ***Timing analysis*** scripts for evaluating performance
  - Per-frame, CPU load

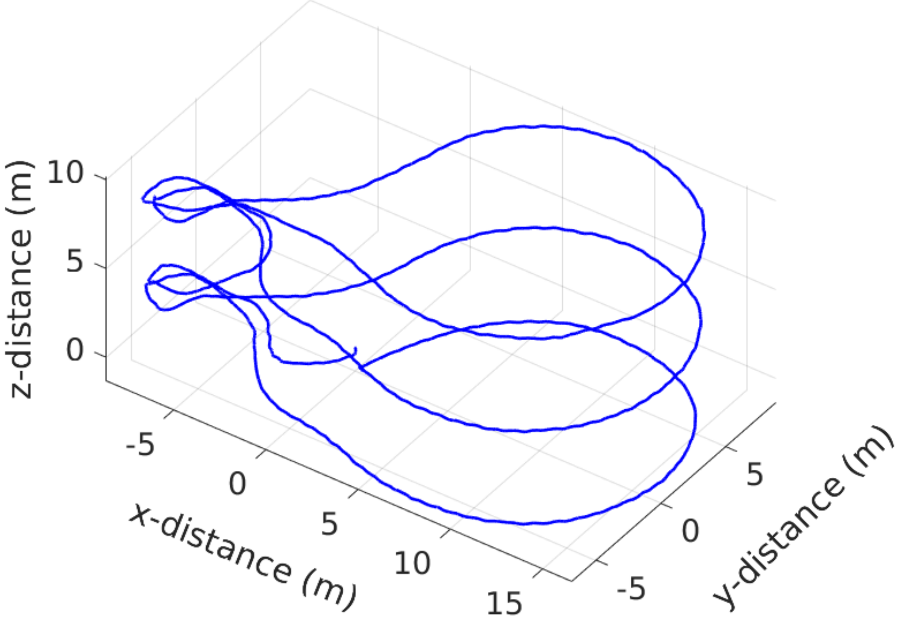


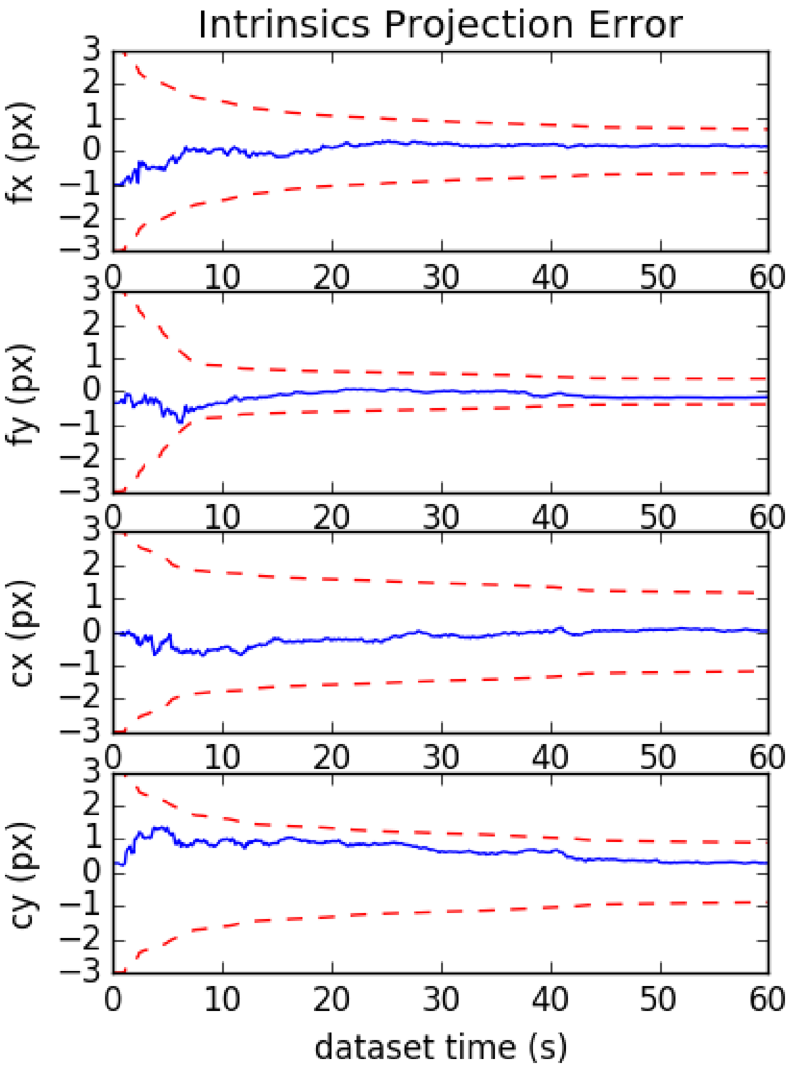
Key Feature – Simulation

- Complete visual-inertial simulation from a given trajectory
- Crucial to algorithm verification



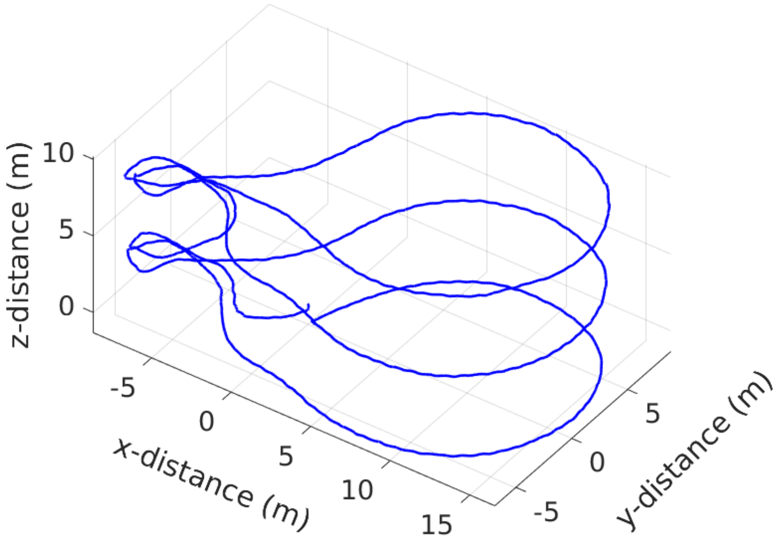
UD Gore Hall





Validation – Calibration

- System able to ***handle poor initial calibration*** and still have low error and be ***consistent***
- If we take the calibration as being ***"true"*** then error quickly grows and estimate becomes ***inconsistent***



Simulated Trajectory

Table 1: Average ATE and NEES over twenty runs with true or bad calibration, with and without online calibration.

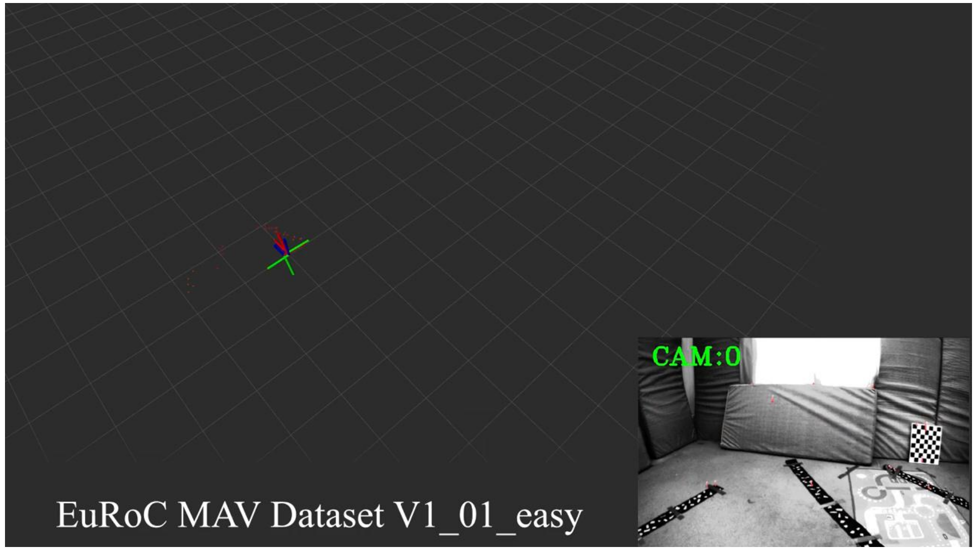
	ATE (deg)	ATE (m)	Ori. NEES	Pos. NEES
true w/ calib	0.212	0.134	2.203	1.880
true w/o calib	0.200	0.128	2.265	1.909
bad w/ calib	0.218	0.139	2.235	2.007
bad w/o calib	5.432	508.719	9.159	1045.174

Validation – Real-world EurocMav Dataset

Table 1: Ten run mean absolute trajectory error (ATE) for each algorithm in units of degree/meters.

	V1_01_easy	V1_02_medium	V1_03_difficult	V2_01_easy	V2_02_medium	Average
mono_ov_slam	0.699 / 0.058	1.675 / 0.076	2.542 / 0.063	0.773 / 0.124	1.538 / 0.074	**1.445** / **0.079**
mono_ov_vio	0.642 / 0.076	1.766 / 0.096	2.391 / 0.344	1.164 / 0.121	1.248 / 0.106	**1.442** / 0.148
mono_okvis	0.823 / 0.090	2.082 / 0.146	4.122 / 0.222	0.826 / 0.117	1.704 / 0.197	1.911 / 0.154
mono_rovioli	2.249 / 0.153	1.635 / 0.131	3.253 / 0.158	1.455 / 0.106	1.678 / 0.153	2.054 / 0.140
mono_rvio	0.994 / 0.094	2.288 / 0.129	1.757 / 0.147	1.735 / 0.144	1.690 / 0.233	1.693 / 0.149
mono_vinsfusion_vio	1.199 / 0.064	3.542 / 0.103	5.934 / 0.202	1.585 / 0.073	2.370 / 0.079	2.926 / **0.104**

- Monocular system with temporal SLAM features able to ***outperform*** state-of-the-art open sourced systems



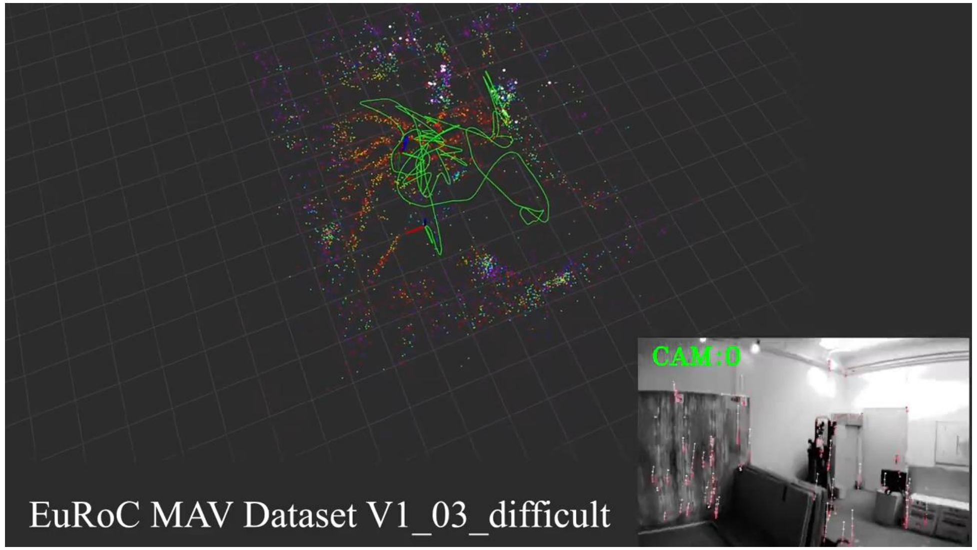
[1] Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W. and Siegwart, R., 2016. The EuRoC micro aerial vehicle datasets. The International Journal of Robotics Research, 35(10), pp.1157-1163.

Validation – Real-world EurocMav Dataset

Table 1: Relative pose error (RPE) for different segment lengths for each algorithm variation over all datasets in units of degree/meters. Note that V2_03 dataset is excluded due the inability for some algorithms to run on it.

	8m	16m	24m	32m	40m	48m
mono_ov_slam	**0.661** / **0.074**	**0.802** / **0.086**	**0.979** / **0.097**	**1.061** / **0.105**	**1.145** / **0.120**	**1.289** / **0.122**
mono_ov_vio	0.826 / 0.094	1.039 / 0.106	1.215 / **0.111**	1.283 / **0.132**	1.342 / **0.151**	1.425 / 0.184
mono_okvis	**0.662** / 0.107	**0.870** / 0.161	1.031 / 0.190	1.225 / 0.213	1.384 / 0.240	1.603 / 0.251
mono_rovioli	1.136 / 0.095	1.585 / 0.135	1.847 / 0.184	2.078 / 0.226	2.218 / 0.263	2.402 / 0.295
mono_rvio	0.705 / 0.130	0.902 / 0.160	**1.029** / 0.183	**1.074** / 0.213	**0.991** / 0.227	**1.077** / 0.232
mono_vinsfusion_vio	0.940 / **0.070**	1.298 / **0.103**	1.680 / 0.118	1.822 / 0.146	1.833 / 0.153	1.860 / **0.171**

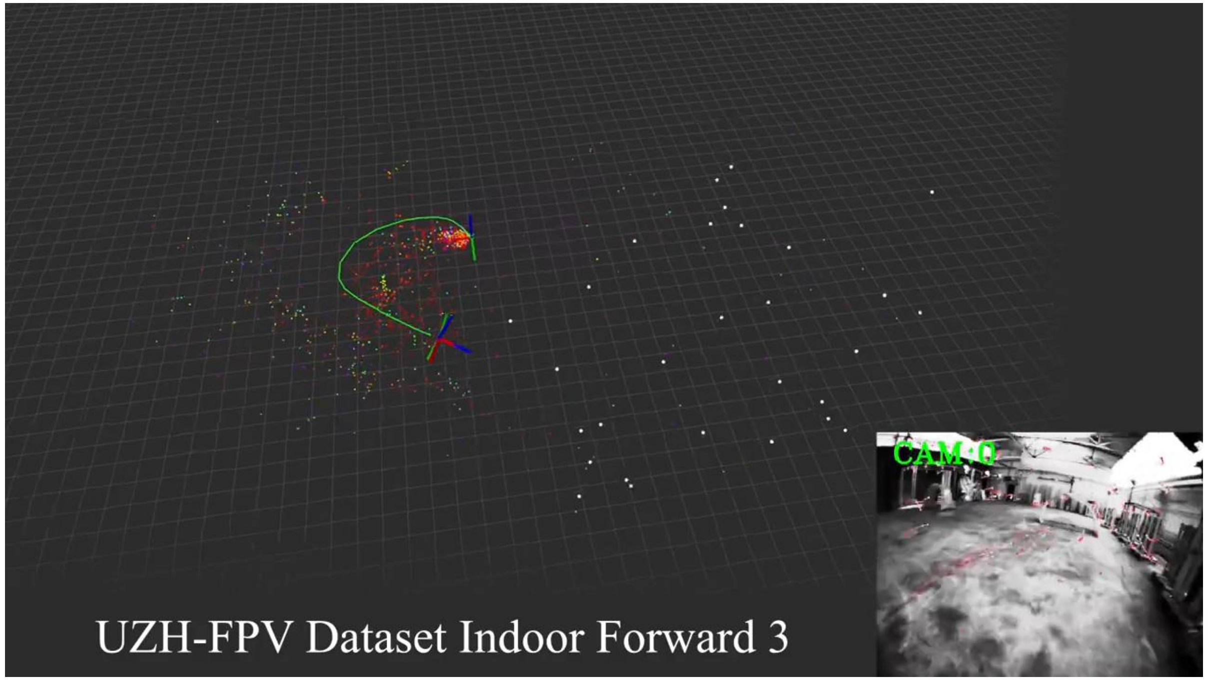
- Relative pose error (RPE) shows improvement over state-of-the-art
- Timing results:**
  - EurocMav VIO: ***4.3x realtime***
  - EurocMav SLAM: ***2.7x realtime***



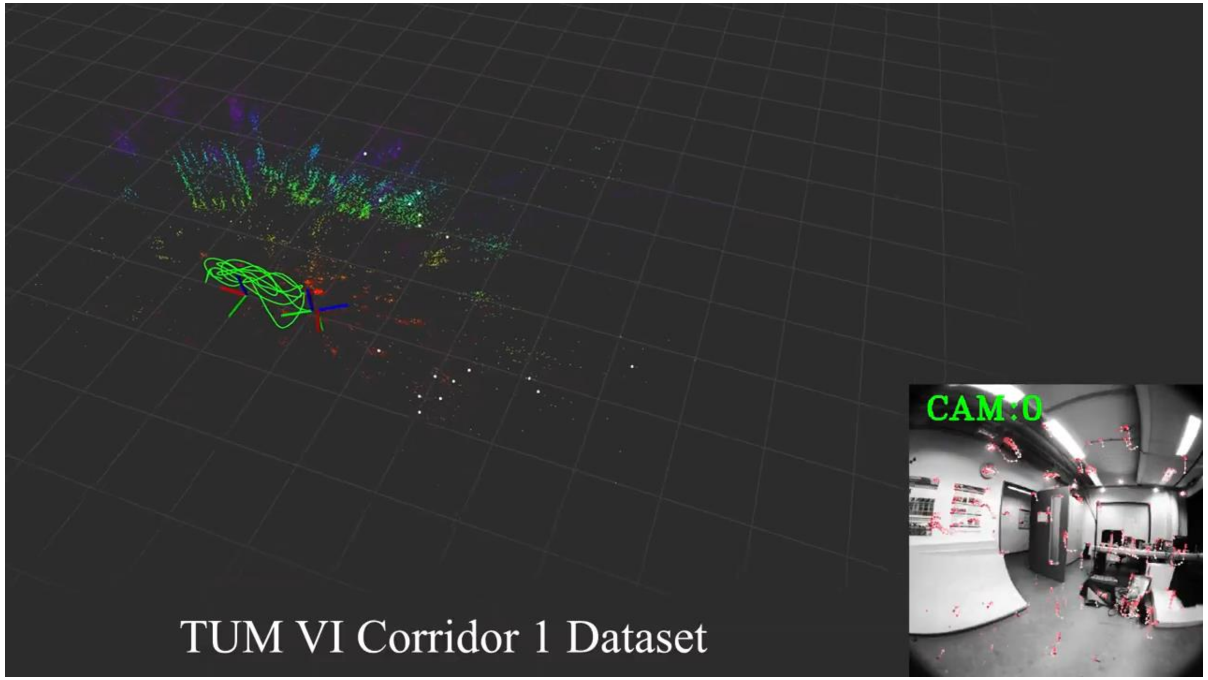
** Timed on an Intel(R) Xeon(R) CPU E3-1505M v6 @3.00GHz processor in single threaded execution

Validation – Other Datasets

UZH-FPV Dataset
High speed > 12.5 m/s
** OpenVINS placed first in 2019 competition **



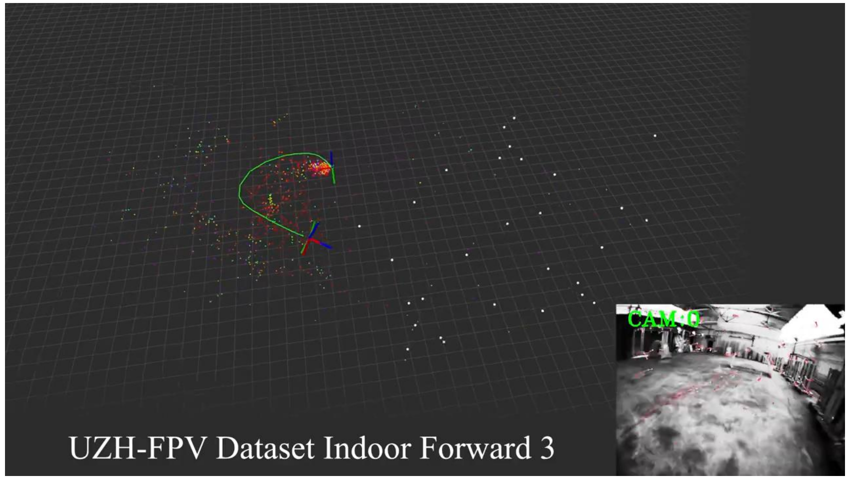
TUM VI Dataset
Indoor Handheld Motion



[1] Delmerico, J., Cieslewski, T., Rebecq, H., Faessler, M. and Scaramuzza, D., 2019, May. Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 6713-6719). IEEE.
[2] Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J. and Cremers, D., 2018, October. The TUM VI benchmark for evaluating visual-inertial odometry. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1680-1687). IEEE.

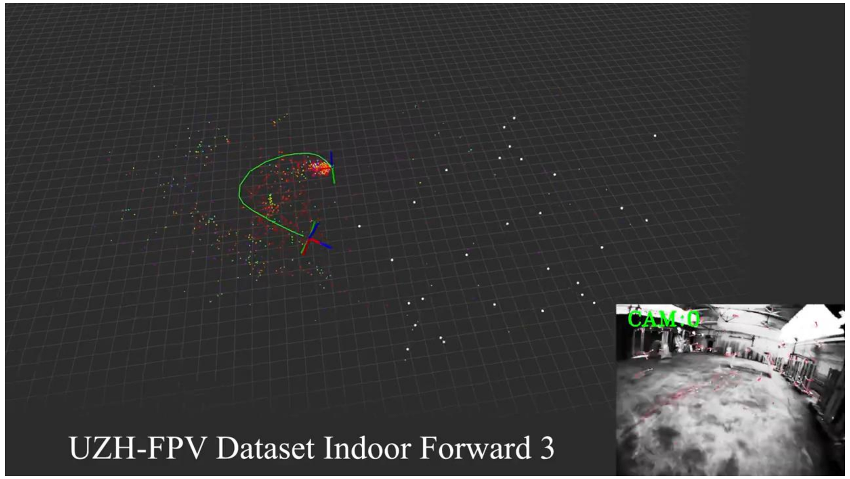
Conclusion

https://github.com/rpng/open_vins
https://docs.openvins.com/



- Presented Work
  - Introduced state-of-the-art open platform (***OpenVINS***) for visual inertial research
  - Support for online camera intrinsic, extrinsic, and time offset calibration
  - Detailed documentation with thorough validation in simulation and realworld experiments

Conclusion

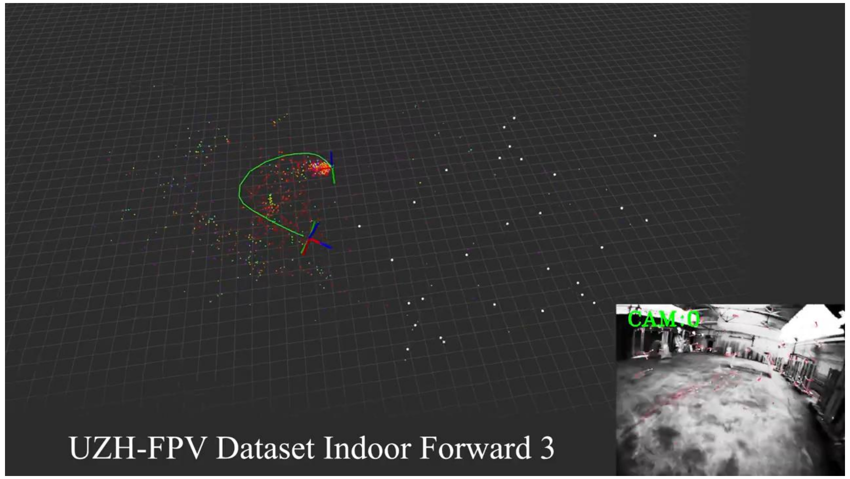


https://github.com/rpng/open_vins
https://docs.openvins.com/

- Presented Work
  - Introduced state-of-the-art open platform (***OpenVINS***) for visual inertial research
  - Support for online camera intrinsic, extrinsic, and time offset calibration
  - Detailed documentation with thorough validation in simulation and realworld experiments

Conclusion

https://github.com/rpng/open_vins
https://docs.openvins.com/



- Presented Work
  - Introduced state-of-the-art open platform (***OpenVINS***) for visual inertial research
  - Support for online camera intrinsic, extrinsic, and time offset calibration
  - Detailed documentation with thorough validation in simulation and realworld experiments

- Current Roadmap
  - DONE: Maplab integration for offline BA (released as ***ov_maplab*** project)
  - DONE: Naïve secondary pose graph (released as ***ov_secondary*** project)
  - FUTURE: Incorporate motion constraints (e.g. zero velocity)
  - FUTURE: Sliding window BA, with SFM initialization