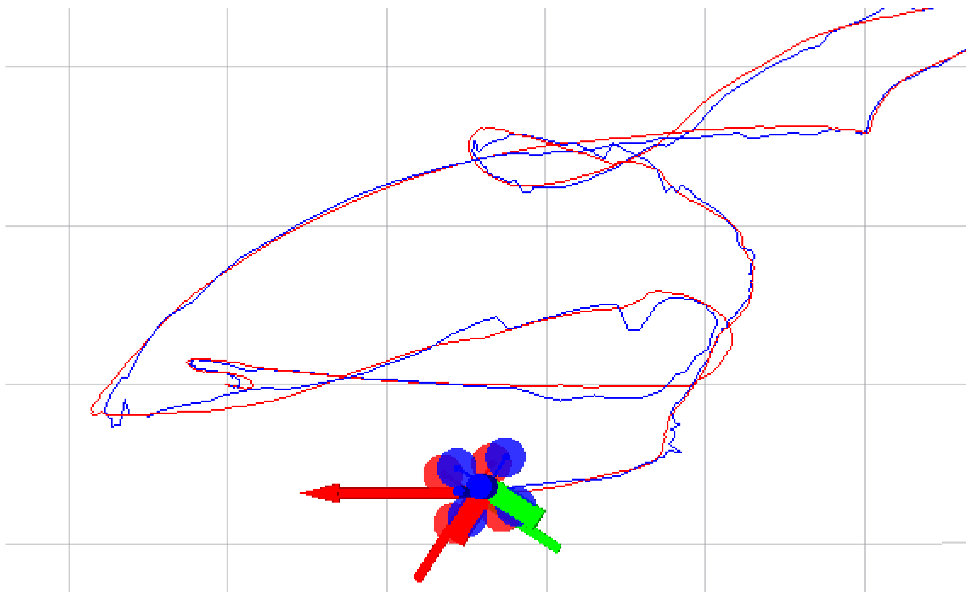
Sensor Fusion for Aerial Robots

Shaojie Shen
Assistant Professor

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Why Sensor Fusion?

- Vision/GPS-only state estimation is too noisy, slow, and delayed for feedback control of agile aerial robots
- To improve robustness with multiple sensors and handle sensor failures
- To estimate quantities that are unobservable using single sensors



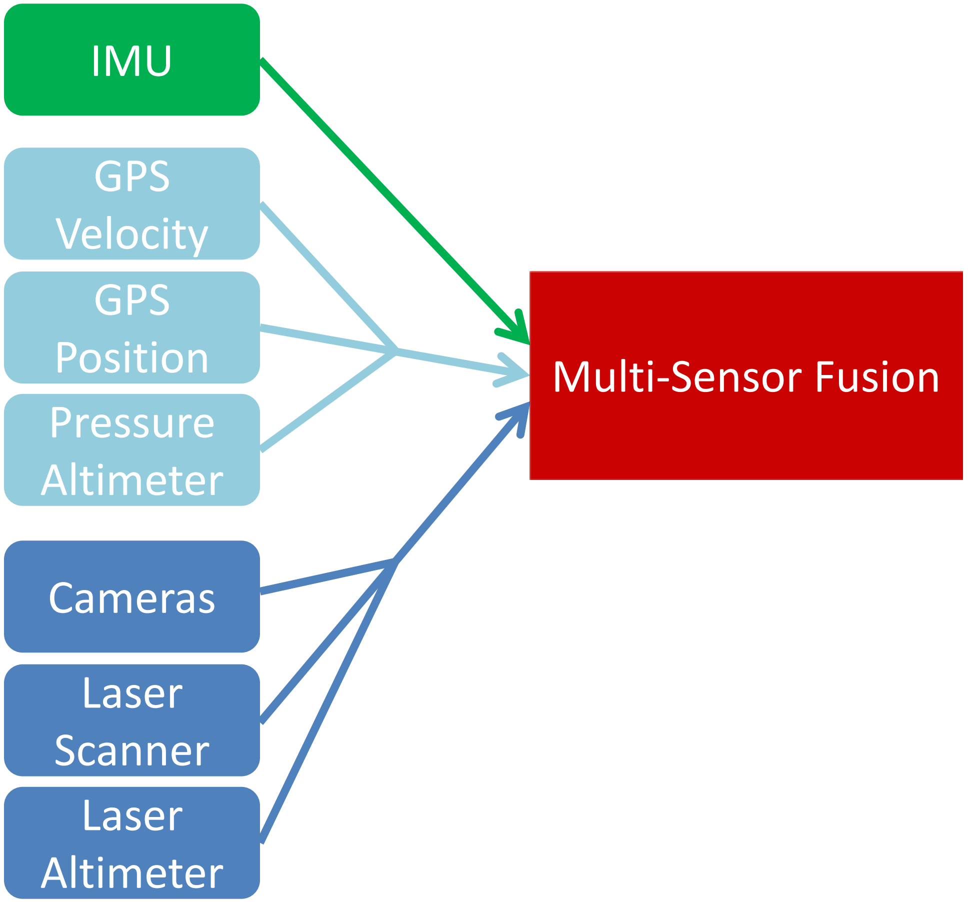
Red: Vision+IMU Fusion
Blue: Vision-only

Design Considerations…

- Accuracy
- Frequency
- Latency
- Sensor synchronization & timestamp accuracy
- Delayed and out-of-order measurements
- Estimator initialization
- Sensor calibration
- Different measurement models with uncertainties
- Robustness to outliers
- Computational efficiency

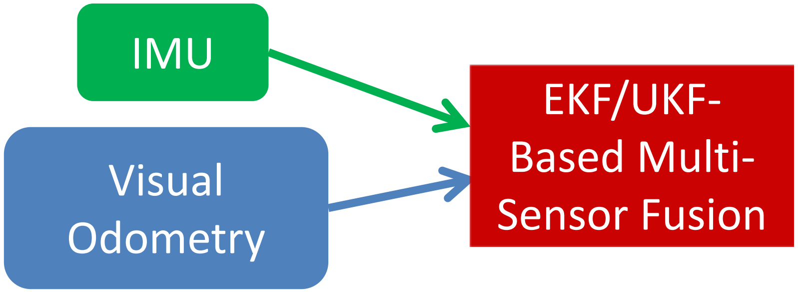
What to Fuse?

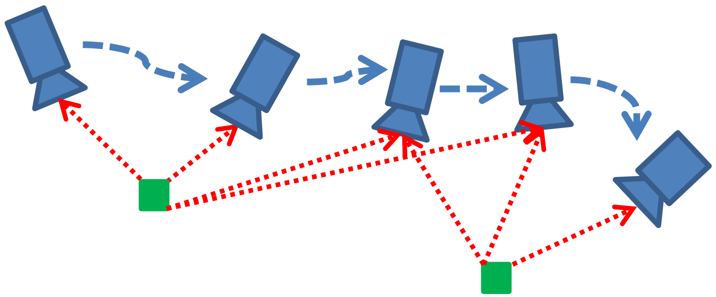
- IMU centric fusion
  - High frequency
  - Low latency
  - (Almost) always available
  - (Usually) large drift
- Absolute measurements
- Relative measurements



(A Few) Sensor Fusion Methods

- Kalman filtering
  - Loosely-coupled [1]
    - Fuse processed information from individual sensors
  - Tightly-coupled [2]
    - Fuse raw measurements directly
- Optimization-based methods
  - Mostly tightly-coupled [3,4]

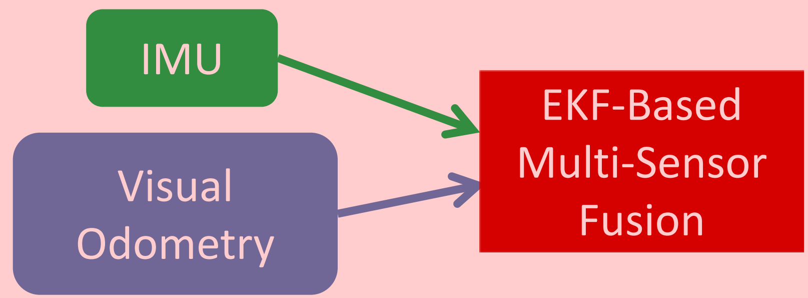




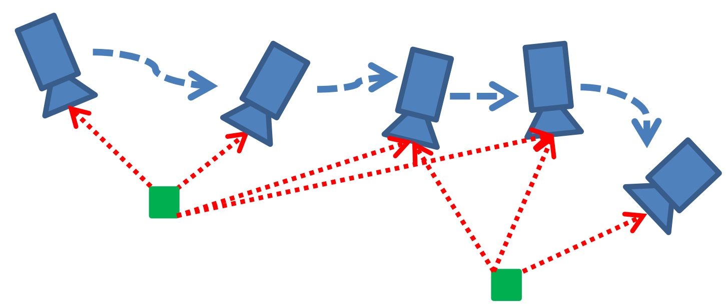
[1] D. Scaramuzza, et. al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS denied environments. IEEE Robot. Autom. Mag., 21(3), 2014.
[2] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proc. of the IEEE Intl. Conf. on Robot. and Autom., pages 3565–3572, Roma, Italy, April 2007.
[3] S. Leutenegger, et al. Keyframe-based visual-inertial SLAM using nonlinear optimization. In Proc. of Robot.: Sci. and Syst., Berlin, Germany, June 2013.
[4] S. Shen, et. al. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," in Proc. of the IEEE Intl. Conf. on Robot. and Autom., Seattle, WA, May 2014.

Outline

- Loosely-Coupled, Extended Kalman Filtering-Based Multi-Sensor Fusion (“Tutorial”)



- Tightly-Coupled, Optimization-Based, Monocular Visual-Inertial Fusion, with Online Initialization and Camera-IMU Extrinsic Calibration (Brief Intro)

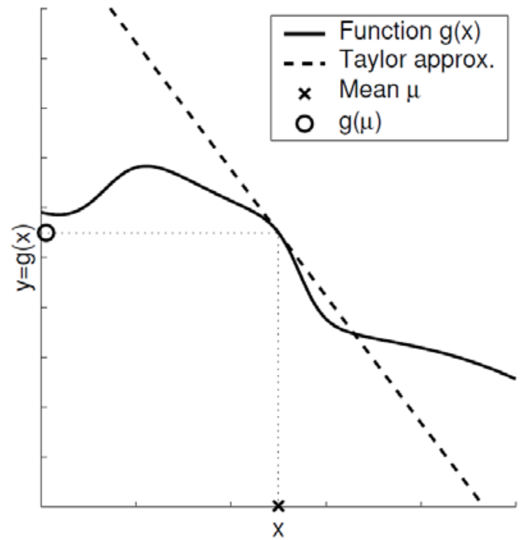


EKF - Assumption & Model

- The prior state of the robot is represented by a Gaussian distribution
  - $p(x_0) \sim N(\mu_0, \Sigma_0)$
- The continuous time process model is:
  - $\dot{x} = f(x, u, n)$
  - $n_t \sim N(0, Q_t)$ is Gaussian white noise
  - Can be linearized using one-step Euler integration:
    - $x_{\bar{t}} \approx x_{t-1} + f(x_{t-1}, u_t, n_t)\, \delta t$
- The measurement model is:
  - $z = h(x, v)$
  - $v_t \sim N(0, R_t)$ is Gaussian white noise
- The process and measurement models are all time-stamped

EKF - Process Model Linearization

- Linearize the process model about $x = \mu_{t-1},\ u = u_t,\ n = 0$
  - $\dot{x} \approx f(\mu_{t-1}, u_t, 0) + \left.\frac{\partial f}{\partial x}\right|_{\mu_{t-1},u_t,0} (x - \mu_{t-1}) + \left.\frac{\partial f}{\partial u}\right|_{\mu_{t-1},u_t,0} (u - u_t) + \left.\frac{\partial f}{\partial n}\right|_{\mu_{t-1},u_t,0} (n - 0)$



- Let:
  - $A_t = \left.\frac{\partial f}{\partial x}\right|_{\mu_{t-1},u_t,0}$
  - $B_t = \left.\frac{\partial f}{\partial u}\right|_{\mu_{t-1},u_t,0}$
  - $U_t = \left.\frac{\partial f}{\partial n}\right|_{\mu_{t-1},u_t,0}$
- Linear process model:
  - $\dot{x} \approx f(\mu_{t-1}, u_t, 0) + A_t(x - \mu_{t-1}) + B_t(u - u_t) + U_t(n - 0)$

EKF – Measurement Model Linearization

- Linearize the measurement model about $x = \bar{\mu}_t,\ v = 0$
  - $h(x, v) \approx h(\bar{\mu}_t, 0) + \left.\frac{\partial h}{\partial x}\right|_{\bar{\mu}_t,0} (x - \bar{\mu}_t) + \left.\frac{\partial h}{\partial v}\right|_{\bar{\mu}_t,0} (v - 0)$
- Let:
  - $C_t = \left.\frac{\partial h}{\partial x}\right|_{\bar{\mu}_t,0}$
  - $W_t = \left.\frac{\partial h}{\partial v}\right|_{\bar{\mu}_t,0}$
- Linear measurement model:
  - $z_t = h(x_t, v_t) \approx h(\bar{\mu}_t, 0) + C_t\,(x_t - \bar{\mu}_t) + W_t\, v_t$

EKF - Summary

- Process Update:
  - $\bar{\mu}_t = \mu_{t-1} + \delta t\, f(\mu_{t-1}, u_t, 0)$
  - $\bar{\Sigma}_t = F_t\, \Sigma_{t-1}\, F_t^T + V_t\, Q_t\, V_t^T$
  - $\dot{x} = f(x, u, n)$
  - $n_t \sim N(0, Q_t)$

  } Assumptions

  - $A_t = \left.\frac{\partial f}{\partial x}\right|_{\mu_{t-1}, u_t, 0}$
  - $U_t = \left.\frac{\partial f}{\partial n}\right|_{\mu_{t-1}, u_t, 0}$

  } Linearization

  - $F_t = I + \delta t\, A_t$
  - $V_t = \delta t\, U_t$

  } Discretization

- Measurement Update:
  - $\mu_t = \bar{\mu}_t + K_t\, (z_t - h(\bar{\mu}_t, 0))$
  - $\Sigma_t = \bar{\Sigma}_t - K_t\, C_t\, \bar{\Sigma}_t$
  - $K_t = \bar{\Sigma}_t\, C_t^T\, (C_t\, \bar{\Sigma}_t\, C_t^T +$

  } Assumptions

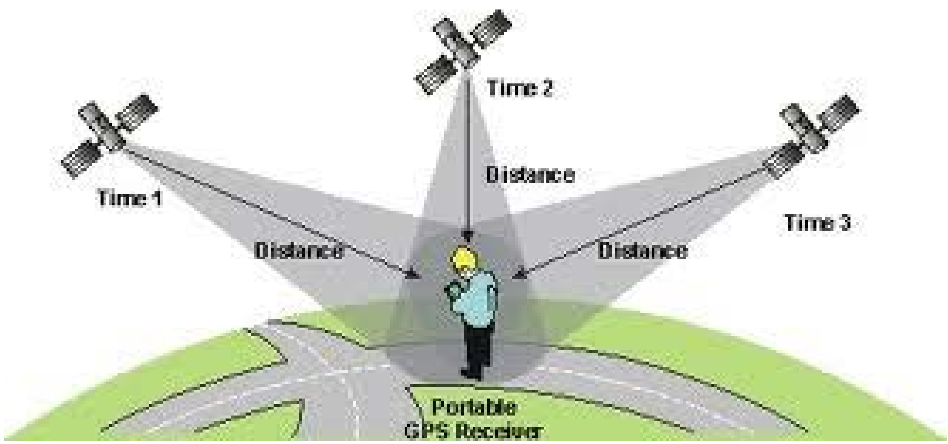
  } Linearization

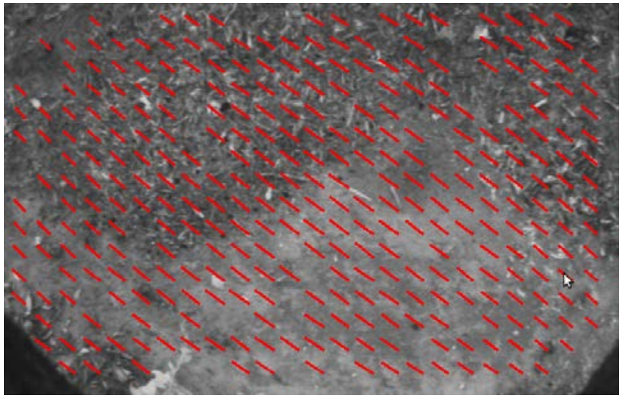
Notes:

- We may have multiple heterogeneous measurement models as long as they are fused sequentially

Example 1

- Quadrotor with IMU and:
  - Absolute Pose Sensors (GPS + Pressure Altimeter + Magnetometer)
  - Absolute Velocity Sensor (Optical Flow / GPS using Doppler effect)





State

- The IMU provides noisy and biased measurements of linear acceleration and angular velocity
- $\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{orientation} \\ \text{linear velocity} \\ \text{gyroscope bias} \\ \text{accelerometer bias} \end{bmatrix} \in \mathbf{R}^{15}$
- (For now) use Z-X-Y Euler angle parameterization of $SO(3)$ for orientation
  - $\mathbf{q} = [\phi, \theta, \psi]^T = [\text{roll}, \text{pitch}, \text{yaw}]^T$
  - $R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$
  - We may use quaternions and error-state EKF to avoid singularities (not covered here)

Process Model

- The gyroscope gives a noisy and biased estimate of the angular velocity
  - $\omega_m = \omega + \mathbf{b}_g + \mathbf{n}_g$
- The drift in the gyroscope bias is a Gaussian, white noise process
  - $\dot{\mathbf{b}}_g = \mathbf{n}_{bg}$
  - $\mathbf{n}_{bg} \sim N(0, Q_g)$
- The accelerometer gives a noisy and baised estimate of the linear acceleration
  - $\mathbf{a}_m = R(\mathbf{q})^T(\ddot{\mathbf{p}} - \mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a$
- The drift in the accelerometer bias is a Gaussian, white noise process
  - $\dot{\mathbf{b}}_a = \mathbf{n}_{ba}$
  - $\mathbf{n}_{ba} \sim N(0, Q_a)$

Process Model

- ω_m is in the body frame, $\mathbf{q}$ is in the world frame
- The angular velocity in the body frame is given by
  - $$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = G(\mathbf{q})\dot{\mathbf{q}}$$
- Process model:
  - $$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ G(\mathbf{q})^{-1}(\omega_m - \mathbf{b}_g - \mathbf{n}_g) \\ \mathbf{g} + R(\mathbf{q})(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a) \\ \mathbf{n}_{bg} \\ \mathbf{n}_{ba} \end{bmatrix}$$

Absolute Measurement Model

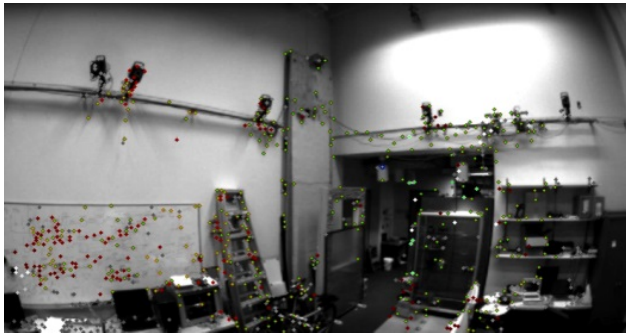
- Position: GPS + Pressure Altimeter
- Orientation: Magnetometer + Accelerometer (not exactly correct…)
- Velocity: Optical flow / GPS with Doppler effect
- $\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \end{bmatrix} + \mathbf{v}$

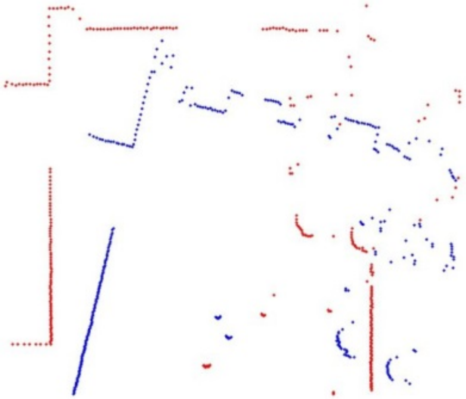
$$= \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{bmatrix} + \mathbf{v}$$

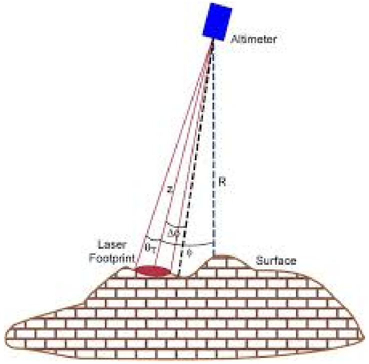
$$= C\,\mathbf{x} + \mathbf{v}$$

Example 2

- Quadrotor with IMU and:
  - 6-DOF Relative Pose Sensor (Stereo Visual Odometry)
  - 3-DOF Relative Pose Sensor (Laser Scan Matching)
  - 1-DOF Relative Pose Sensor (Laser/ Sonar Altimeter)
- These are all measurements with respect to a certain keyframe

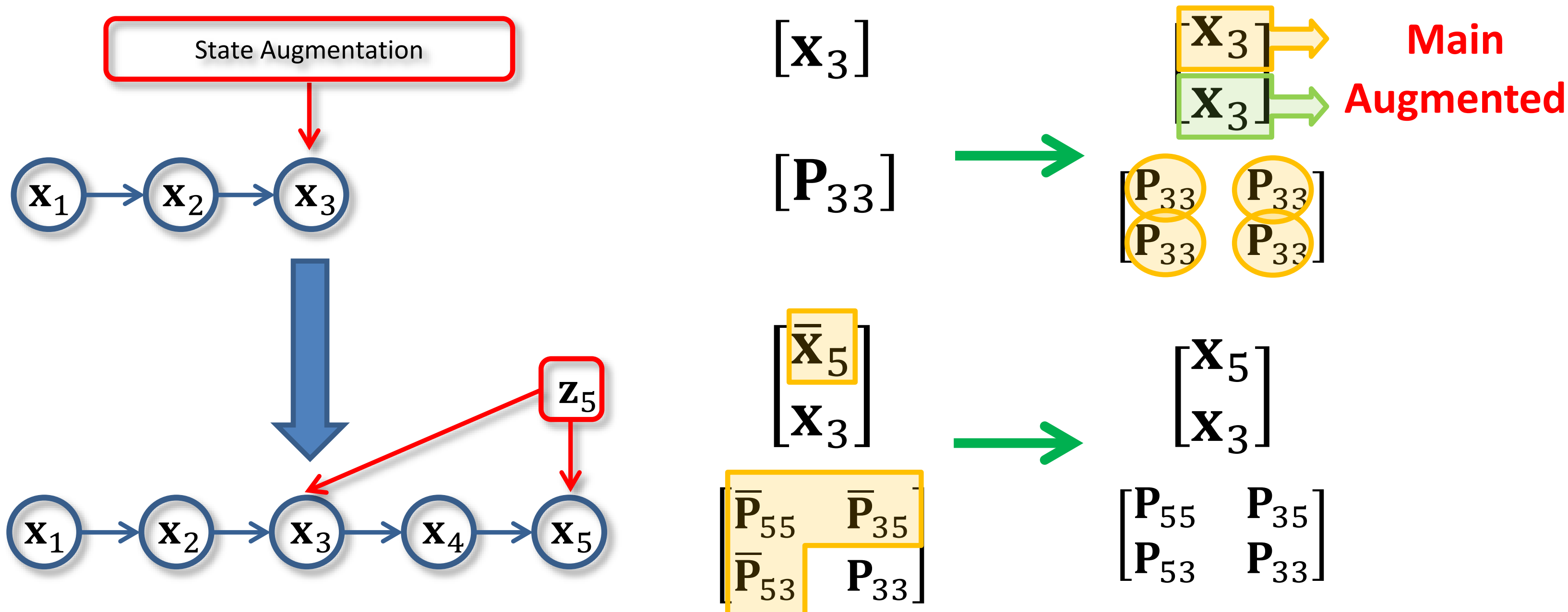






State Augmentation

- Kalman filter requires all measurements to be related to the current state only
- State augmentation at keyframe changes (Roumeliotis and Burdick, 2002)
  - May augment arbitrary copies of states depending on the availability of relative sensors



Relative Measurement Models

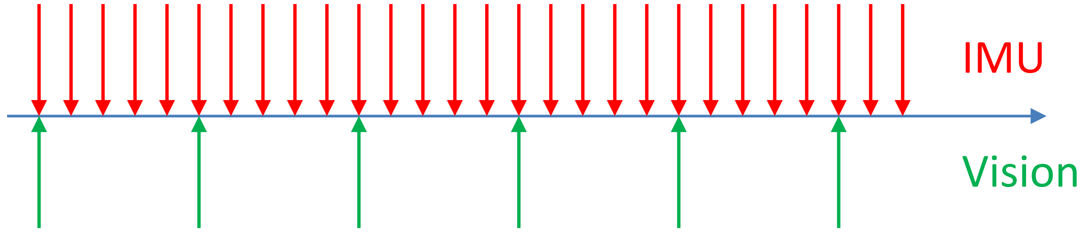
- 6-DOF relative pose measurement from stereo visual odometry
- Both $\mathbf{x}_t$ and $\mathbf{x}_{t-k}$ are augmented into a joint state $\mathbf{y}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-k} \end{bmatrix}$
- $$\begin{aligned} \mathbf{z} &= \begin{bmatrix} \Delta\mathbf{p}_{t|t-k} \\ \Delta\mathbf{q}_{t|t-k} \end{bmatrix} + \mathbf{v}_\mathrm{t} \\ &= \begin{bmatrix} R(\mathbf{q}_{t-k})(\mathbf{p}_t - \mathbf{p}_{t-k}) \\ toEuler(R(\mathbf{q}_{t-k})^T R(\mathbf{q}_t)) \end{bmatrix} + \mathbf{v}_\mathrm{t} \\ &= h(\mathbf{x}_t, \mathbf{x}_{t-k}) + \mathbf{v}_\mathrm{t} \\ &= h(\mathbf{y}_t) + \mathbf{v}_\mathrm{t} \end{aligned}$$

Notes:

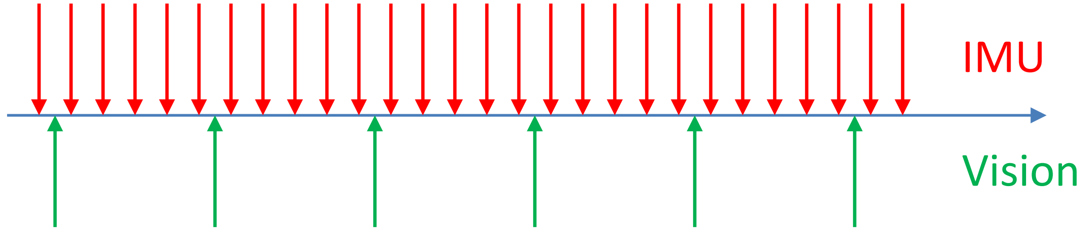
- One state augmentation per keyframe change
- Only states that are affected by the relative measurement (pose in the example) need to be augmented

Sensor synchronization & timestamps

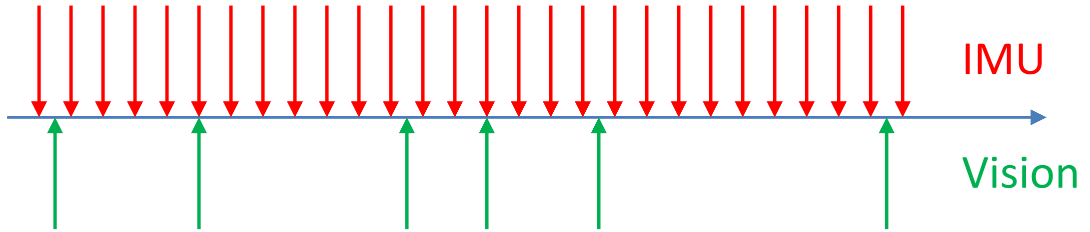
- Best: Sensors perfectly synchronized



- OK: Sensors have the same clock

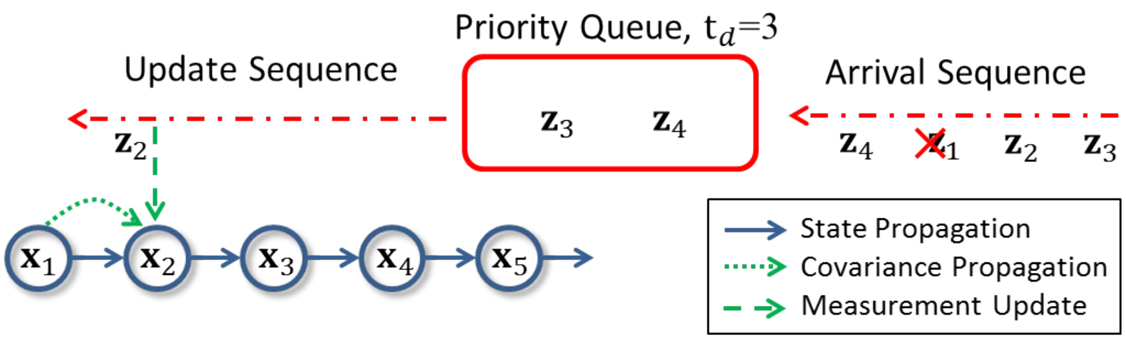


- Bad: Sensors with different clock or inaccurate timestamps



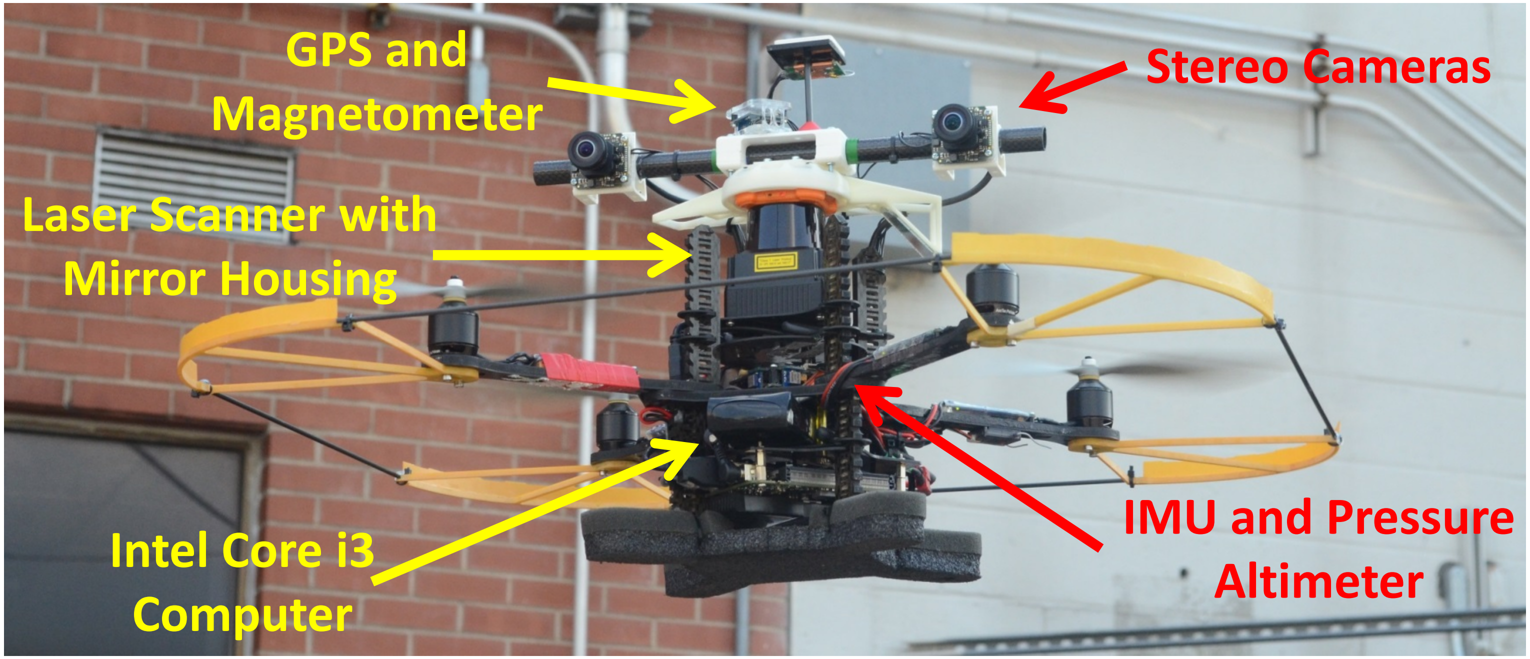
Delayed and Out-of-Sequence Measurements

- Handles delayed and out-of-sequence measurements using fixed-lag priority queue
  - Redo process process updates (state only) after each (delayed) measurement update
  - Covariance update is done only up to the latest measurement



Multi-Sensor Quadrotor

- UKF-based multi-sensor fusion

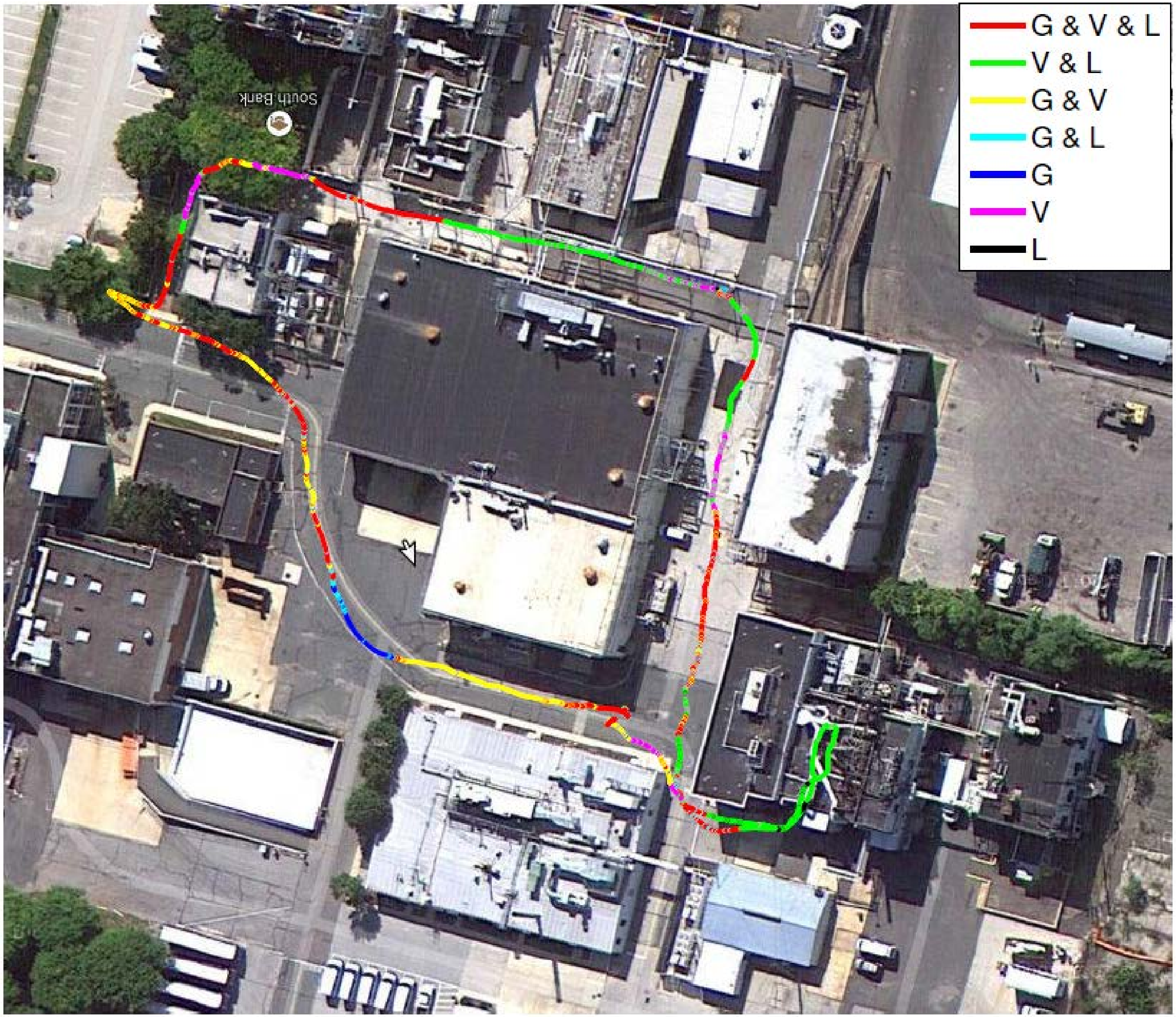


Sensors: IMU, Laser, Cameras, GPS

Autonomous Flight

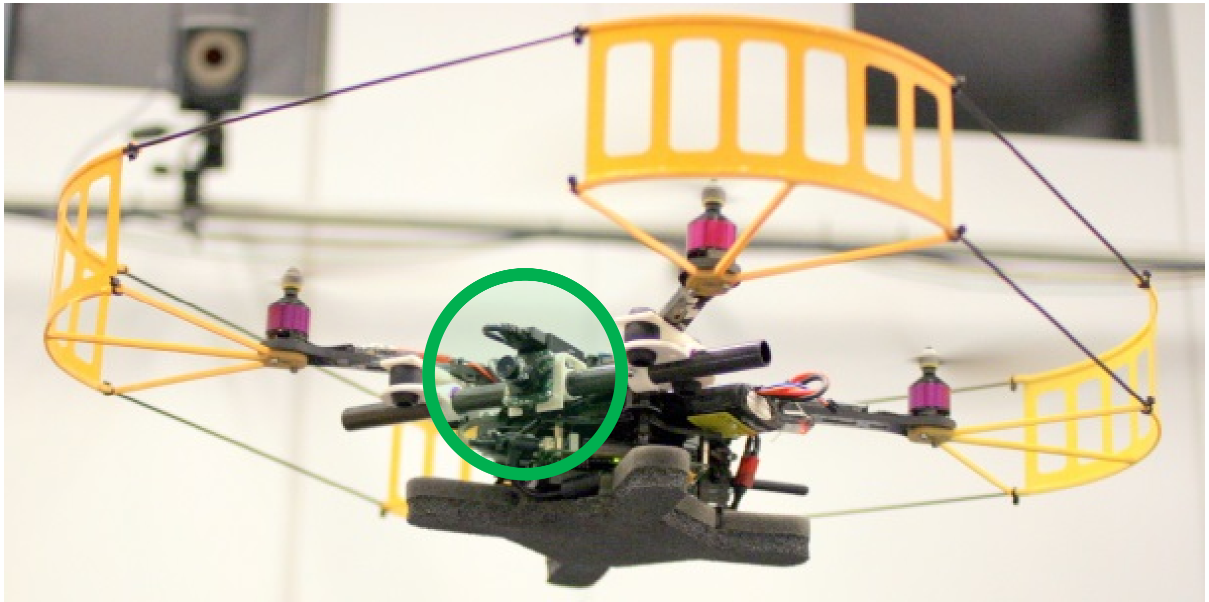
All Processing Onboard

Length: 450 m, Speed: 1.5m/s


G & V & L
V & L
G & V
G & L
G
V
L

Example 3

- Quadrotor with IMU and:
  - Up-to-Scale Relative Pose Sensor (Monocular Visual Odometry)
  - Very relevant for micro aerial robots!



Loosely-Coupled Monocular Visual-Inertial Fusion

- State is modified to explicitly estimate the visual scale:
  - $$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_a \\ \lambda \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{orientation} \\ \text{linear velocity} \\ \text{gyroscope bias} \\ \text{accelerometer bias} \\ \text{visual scale} \end{bmatrix} \in \mathbf{R}^{16}$$

May suffer from convergence issues if scale initialization is poor

- Relative measurement model ($\Delta\check{\mathbf{p}}$: up-to-scale relative translation):
  - $$\begin{aligned} \mathbf{z} &= \begin{bmatrix} \Delta\check{\mathbf{p}}_{t|t-k} \\ \Delta\mathbf{q}_{t|t-k} \end{bmatrix} + \mathbf{v}_\mathrm{t} \\ &= \begin{bmatrix} \lambda \cdot R(\mathbf{q}_{t-k})(\mathbf{p}_t - \mathbf{p}_{t-k}) \\ toEuler(R(\mathbf{q}_{t-k})^T R(\mathbf{q}_t)) \end{bmatrix} + \mathbf{v}_\mathrm{t} \\ &= h(\mathbf{x}_t, \mathbf{x}_{t-k}) + \mathbf{v}_\mathrm{t} \end{aligned}$$

Summary of Loosely-Coupled EKF Fusion

- Accuracy 😐
- Frequency ☺
- Latency ☺
- ~~Sensor synchronization & timestamp accuracy ☹~~
  - Can be done using offline calibration if sensors have the same clock
- Delayed and out-of-order measurements ☺
- Estimator initialization ☹
- Sensor calibration 😐
  - Can be done using offline calibration and online refinement
- Different measurement models with uncertainties ☺
- Robustness to outliers 😐
- Computational efficiency ☺

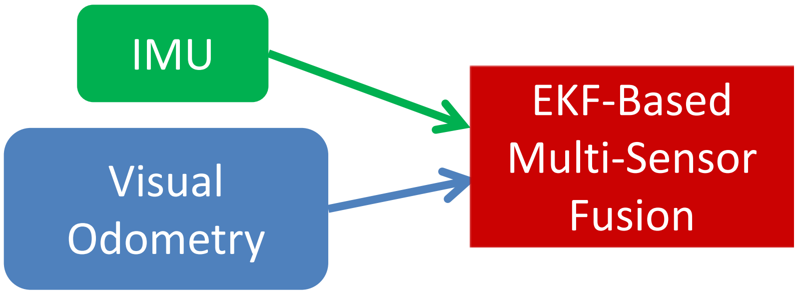
Open Source Packages

- Kalibr - Offline calibration toolbox (ETH Zurich ASL)
  - Multi-camera calibration
  - Camera-IMU calibration
  - Temporal alignment: handles sensor synchronization & timestamp accuracy issues
  - https://github.com/ethz-asl/kalibr

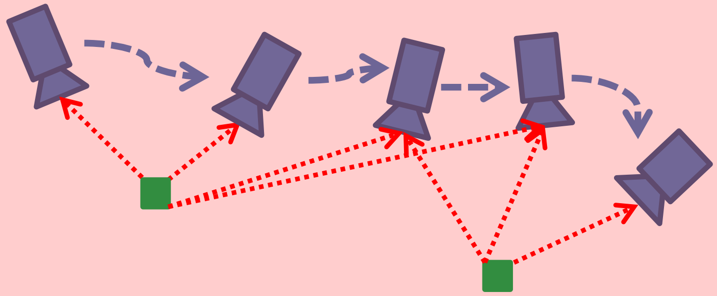
- MSF - Modular framework for multi sensor fusion based on an EKF (ETH Zurich ASL)
  - https://github.com/ethz-asl/ethzasl_msf

Outline

- Loosely-Coupled, Extended Kalman Filtering-Based Multi-Sensor Fusion ("Tutorial")

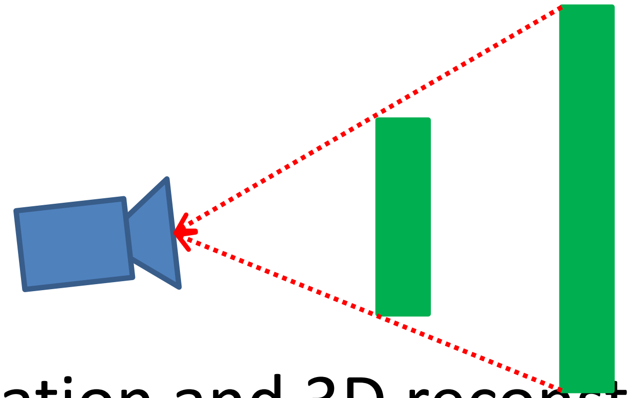


- Tightly-Coupled, Optimization-Based, Monocular Visual-Inertial Fusion, with Online Initialization and Camera-IMU Extrinsic Calibration (Brief Intro)

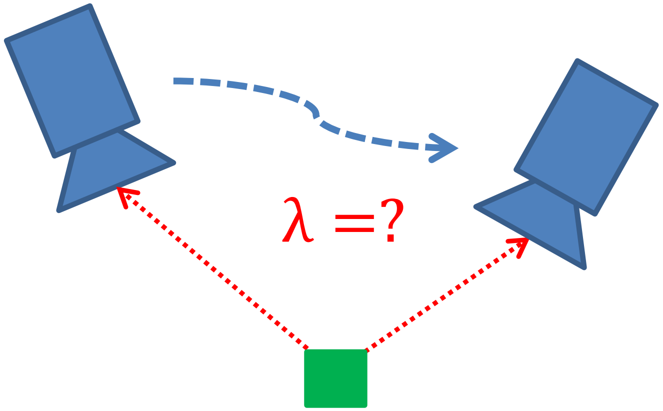


Challenges

- Scale ambiguity



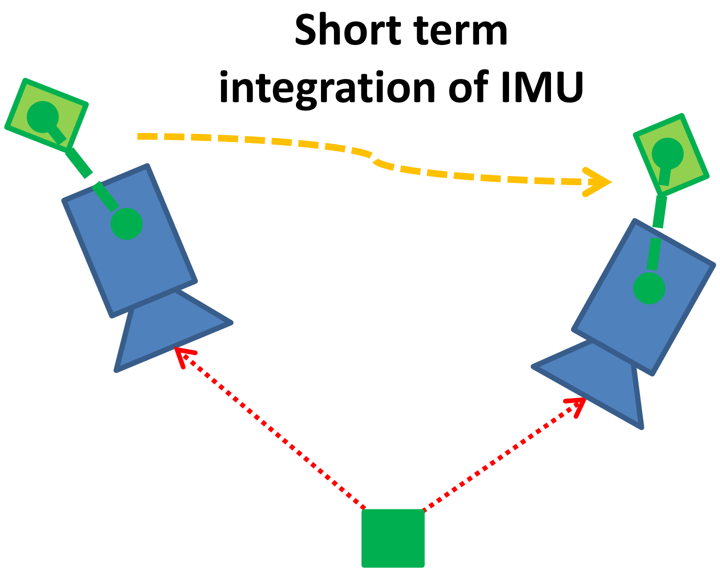
- Up-to-scale motion estimation and 3D reconstruction (Structure from Motion)



Challenges

- With IMU, scale is observable, but…
  - Requires initial velocity and attitude (gravity)
  - Requires knowledge about camera-IMU calibration
  - Highly nonlinear system – requires initial values to converge

$v_0 = ?$
$g_0 = ?$
$R_c^b = ?$
$p_c^b = ?$

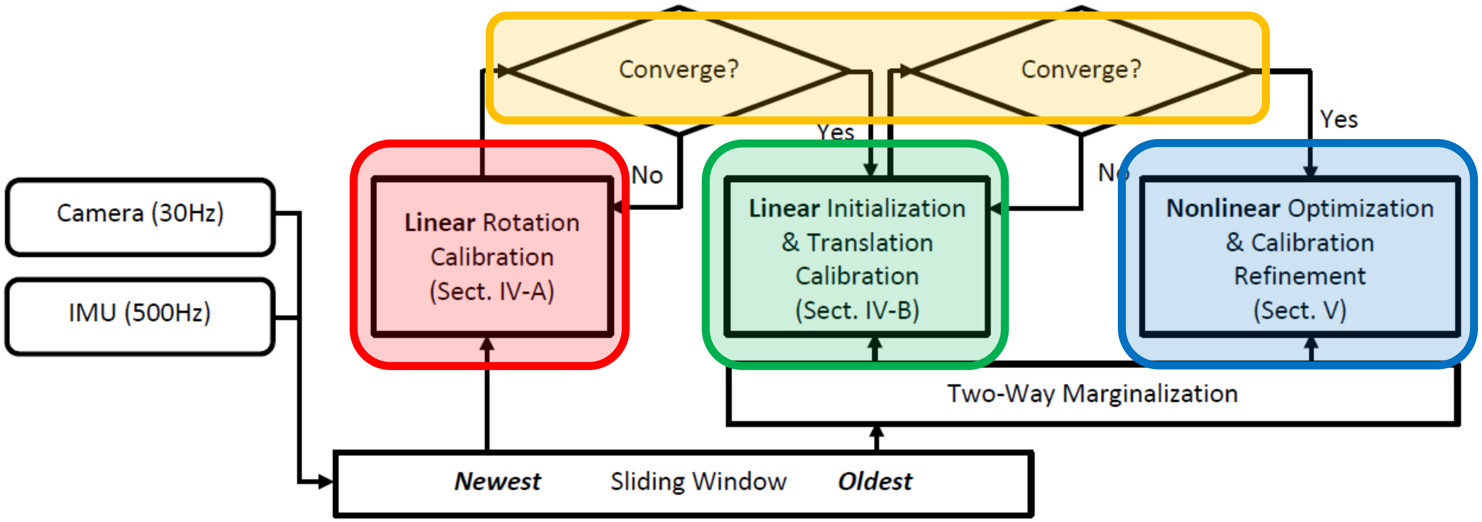


Can we operate without initialization or calibration?

Plug-and-play monocular visual-inertial systems!

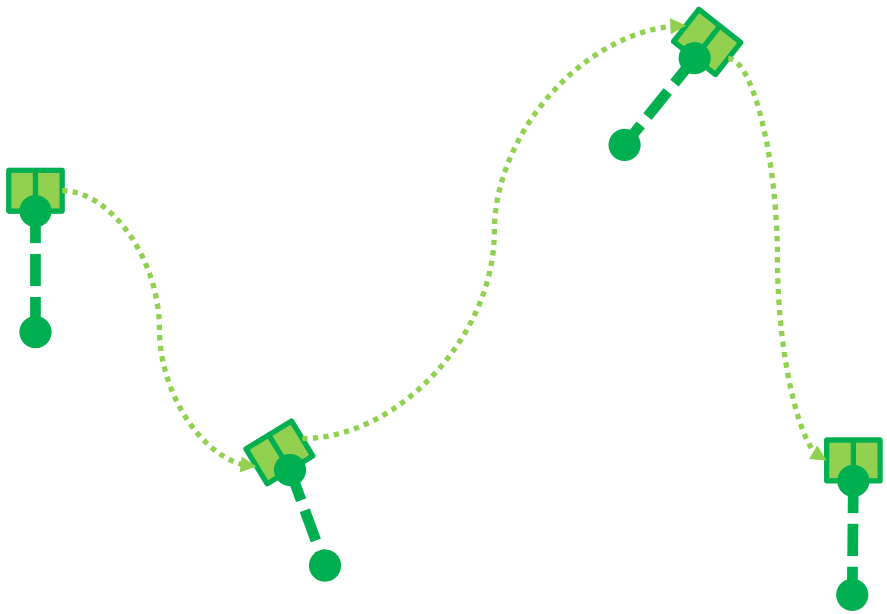
System Pipeline

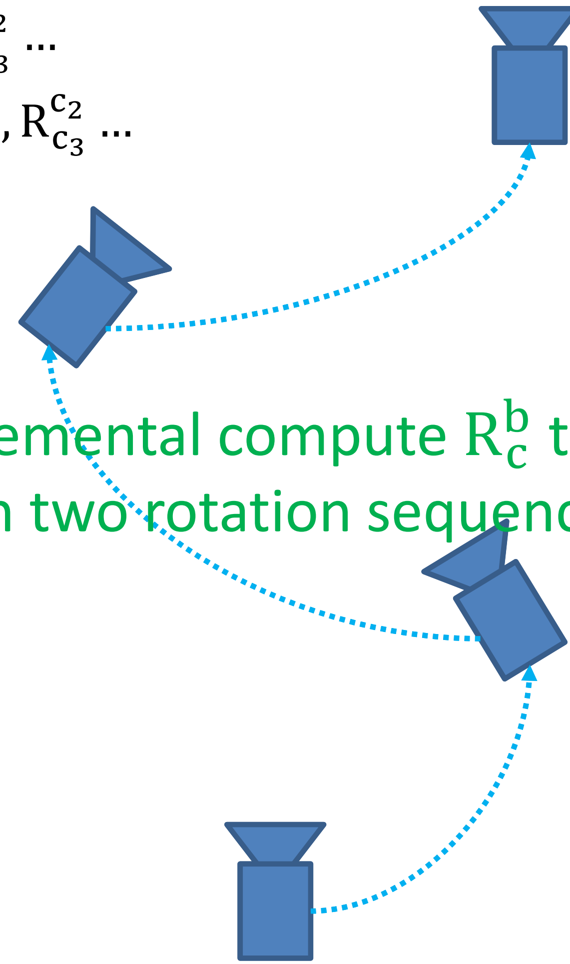
- Linear camera-IMU rotation calibration (R_c^b)
- Linear initialization and camera-IMU translation calibration (v_0, g_0, p_c^b)
- Tightly-coupled nonlinear optimization and calibration refinement
- Calibration convergence identification methods



Linear Camera-IMU Rotation Calibration

- Incremental IMU Rotations: $R_{b_1}^{b_0}, R_{b_2}^{b_1}, R_{b_3}^{b_2}$ …
- Incremental Camera Rotations: $R_{c_1}^{c_0}, R_{c_2}^{c_1}, R_{c_3}^{c_2}$ …





Incremental compute R_c^b to align two rotation sequences

IMU Pre-Integration

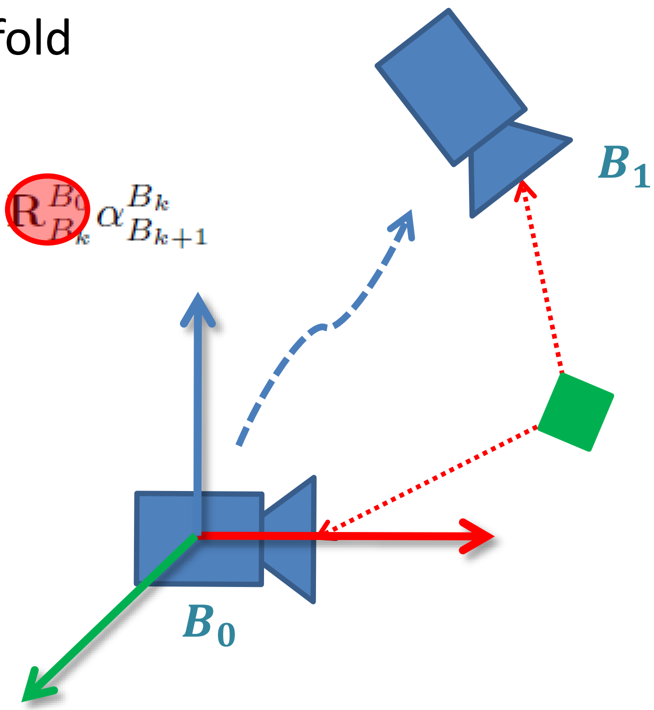
- IMU integration in the body frame of the first pose
  - Nonlinearity from relative rotation only
  - Linear update equations for position, velocity, and gravity
  - IMU Integration without initialization
  - Uncertainty propagation on manifold

$$\mathbf{p}^{B_0}_{B_{k+1}} = \mathbf{p}^{B_0}_{B_k} + \mathbf{v}^{B_0}_{B_k}\Delta t - \mathbf{g}^{B_0}\Delta t^2/2 + \mathbf{R}^{B_0}_{B_k}\alpha^{B_k}_{B_{k+1}}$$

$$\mathbf{v}^{B_0}_{B_{k+1}} = \mathbf{v}^{B_0}_{B_k} - \mathbf{g}^{B_0}\Delta t + \mathbf{R}^{B_0}_{B_k}\beta^{B_k}_{B_{k+1}}$$

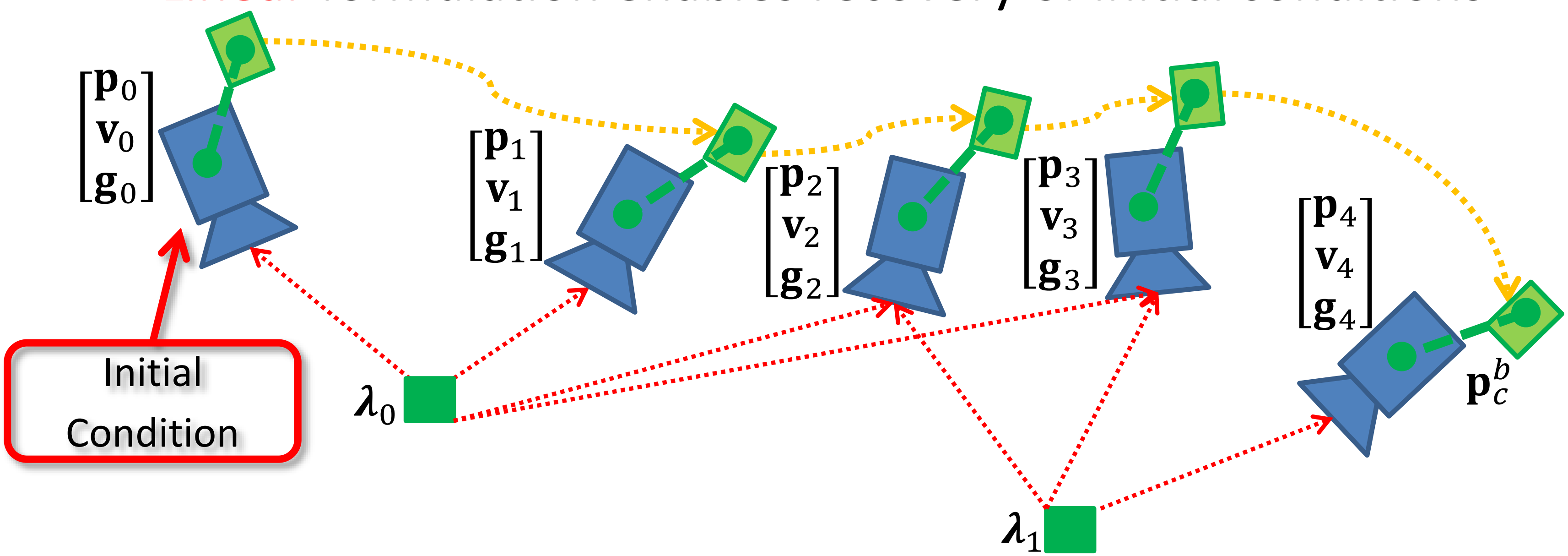
$$\alpha^{B_k}_{B_{k+1}} = \iint_{B_k}^{B_{k+1}} \mathbf{R}^{B_k}_{B}\mathbf{a}^{B}dt^2$$

$$\beta^{B_k}_{B_{k+1}} = \int_{B_k}^{B_{k+1}} \mathbf{R}^{B_k}_{B}\mathbf{a}^{B}dt$$

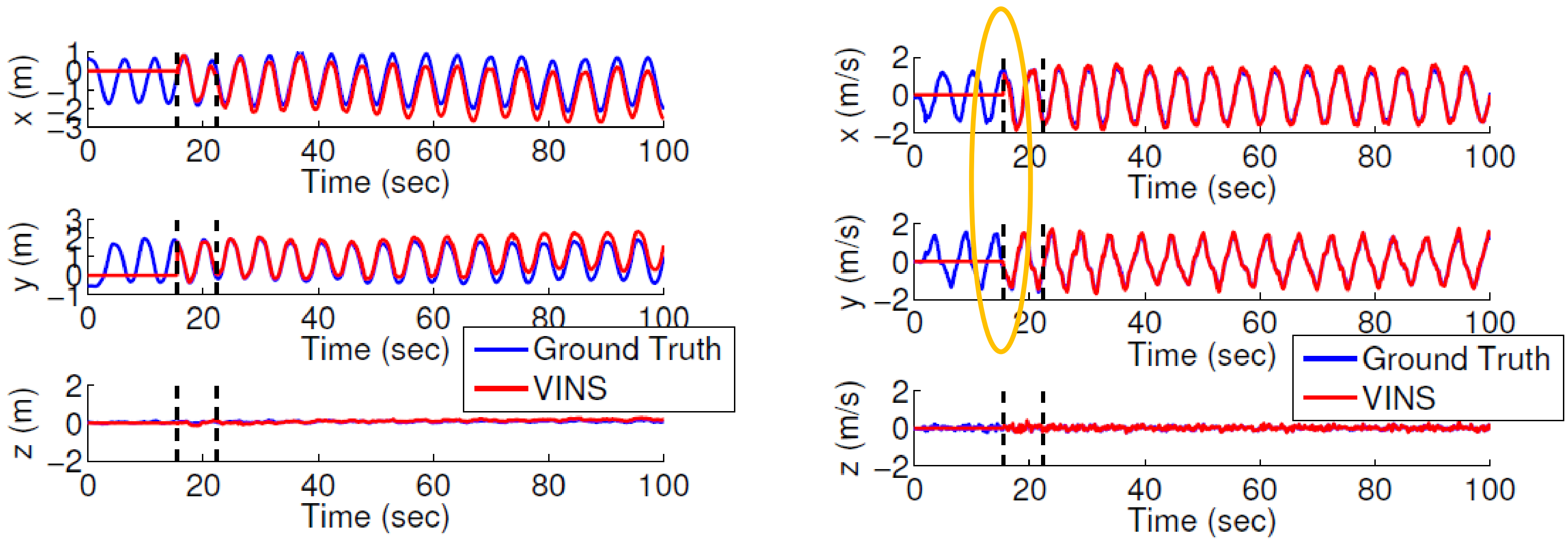


Linear Sliding Window Initialization and Camera-IMU Translation Calibration

- IMU pre-integration for using IMU without initialization
- Estimates position, velocity, gravity, camera-IMU translation, and feature depth using graph-based optimization
- Linear formulation enables recovery of initial conditions

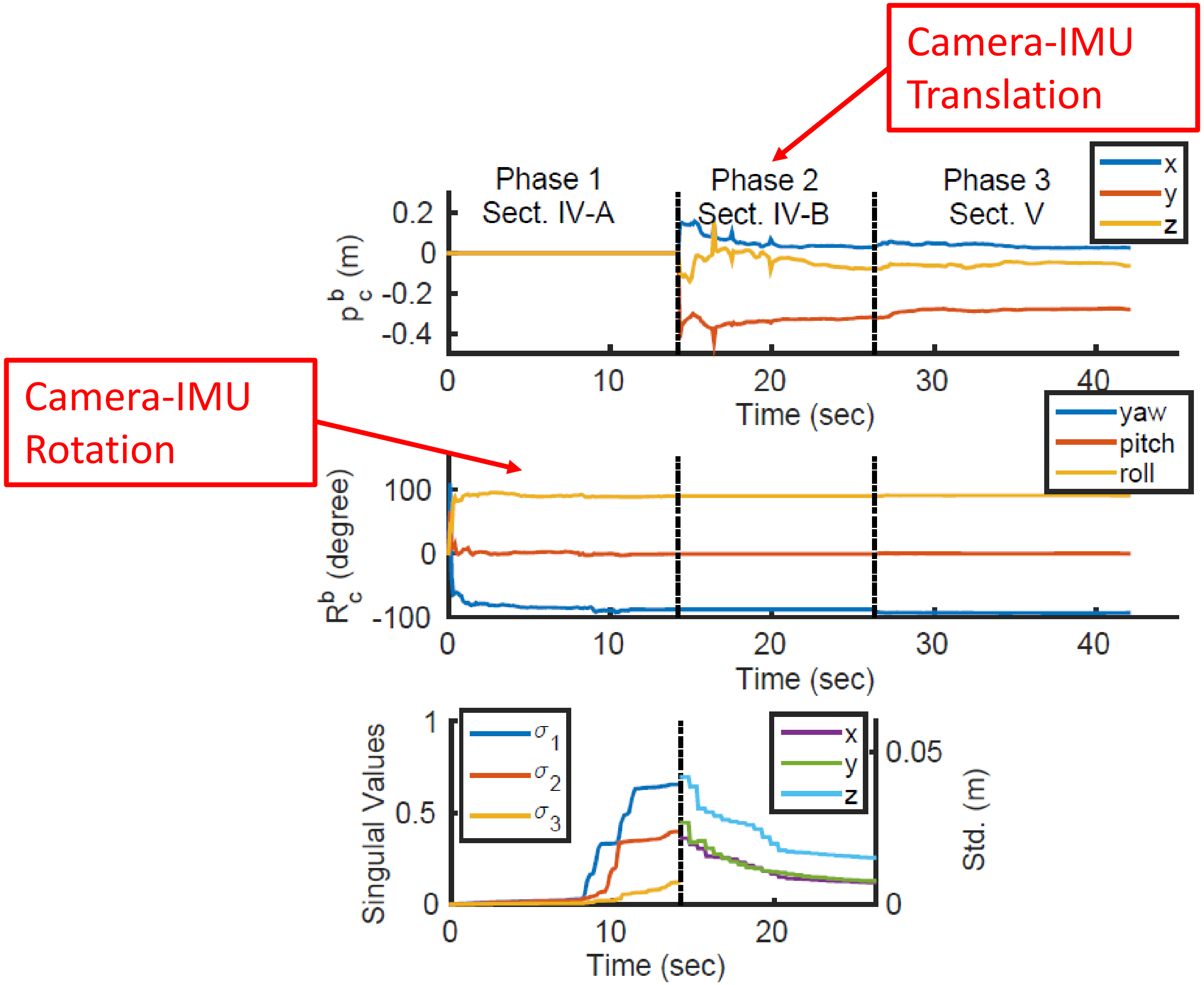


On-the-Fly Initialization

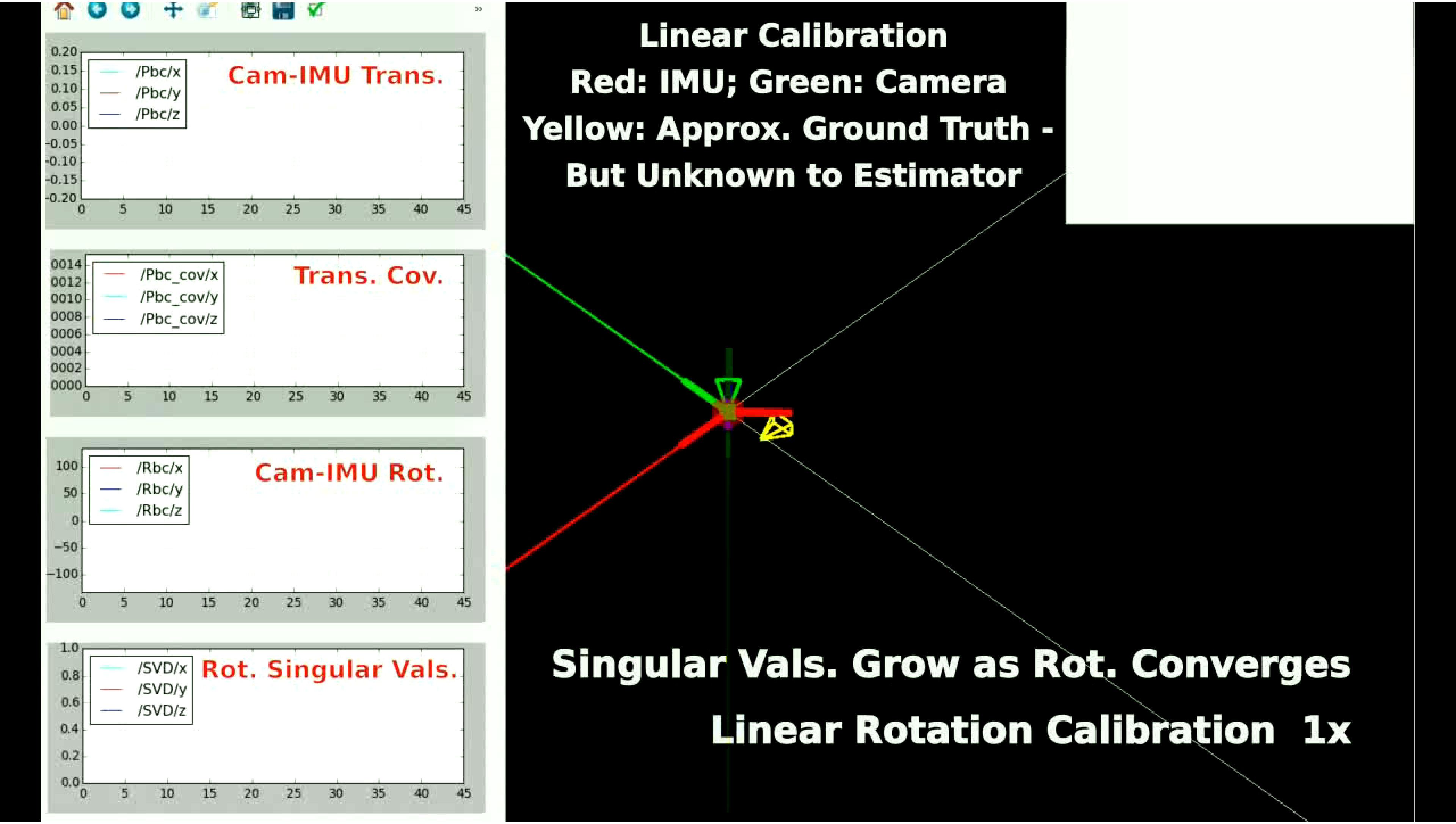


(a) Position (b) Velocity

Camera-IMU Calibration

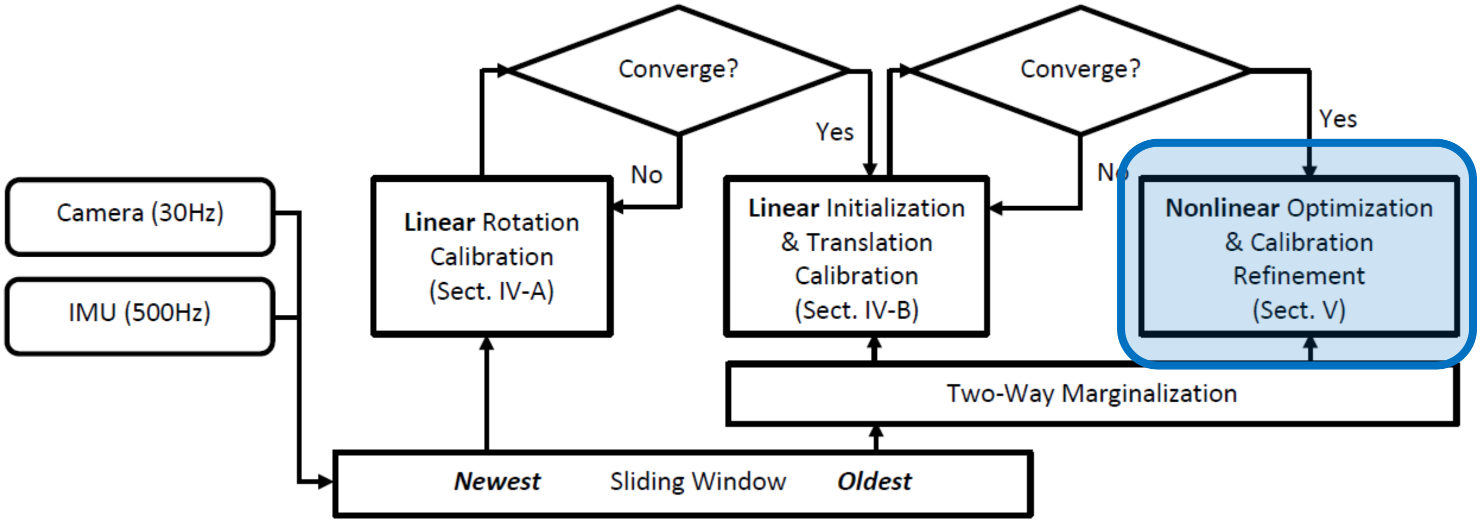


Linear Initialization and Camera-IMU Calibration



System Pipeline

- Tightly-coupled nonlinear optimization and calibration refinement



Tightly-Coupled Nonlinear Sliding Window Optimization with Calibration Refinement

- Nonlinear graph-based optimization using linear initialization
  - Optimize position, velocity, rotation, inverse feature depth, and camera-IMU transformation simultaneously:

$$\mathcal{X} = \left[\mathbf{x}_n,\, \mathbf{x}_{n+1},\, \cdots\, \mathbf{x}_{n+N},\, \mathbf{x}_c^b,\, \lambda_0,\, \lambda_{m+1},\, \cdots\, \lambda_{m+M}\right]$$

$$\mathbf{x}_k = \left[\mathbf{p}_{b_k}^{b_0},\, \mathbf{v}^{b_k},\, \mathbf{q}_{b_k}^{b_0}\right]$$

$$\mathbf{x}_c^b = \left[\mathbf{p}_c^b,\, \mathbf{q}_c^b\right]$$

  - Iteratively minimize residuals from all sensors

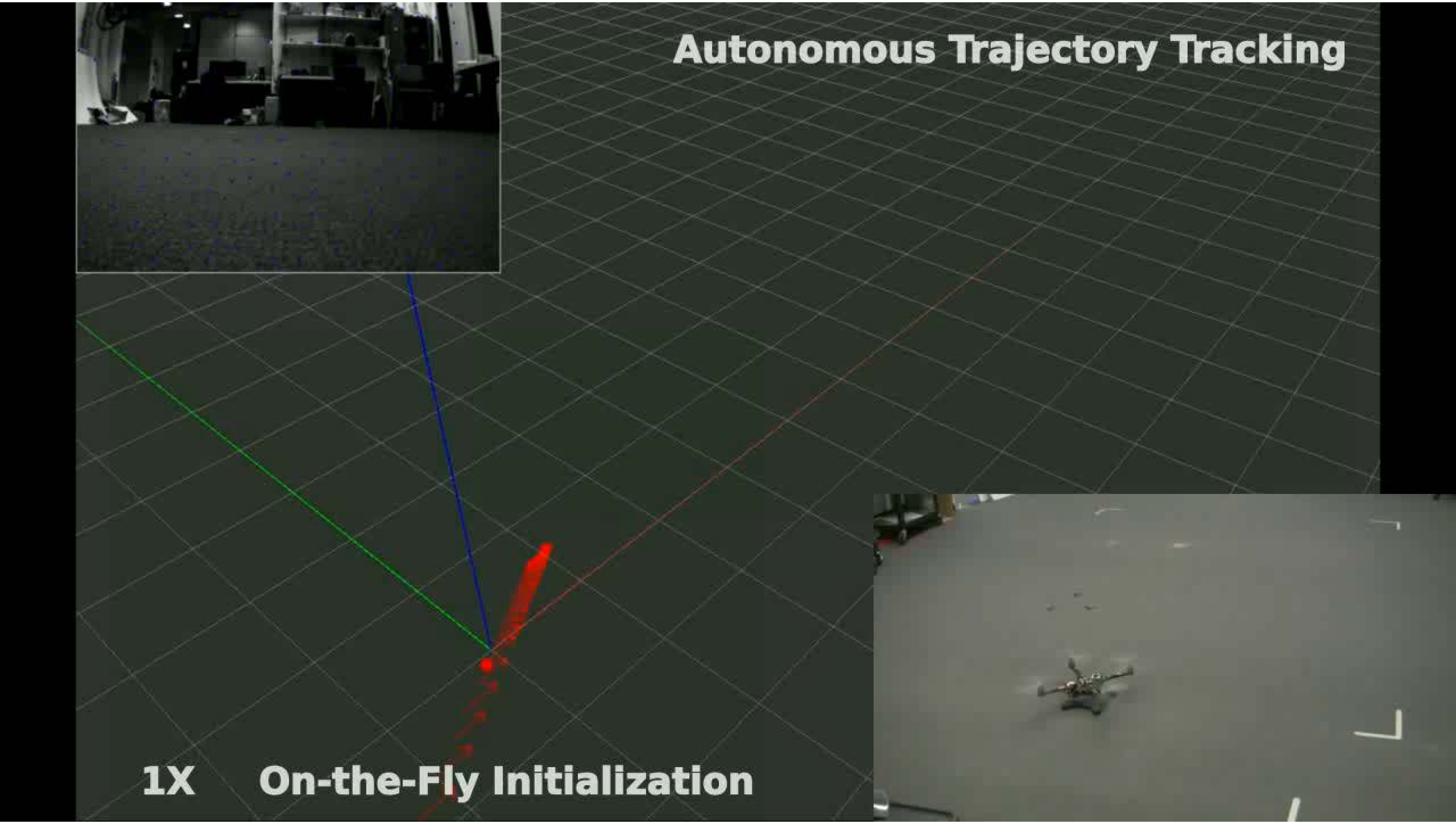
$$\min_{\mathcal{X}} \left\{ (\mathbf{b}_p - \mathbf{\Lambda}_p \mathcal{X}) + \sum_{k \in \mathcal{D}} \left\| r_{\mathcal{D}}(\hat{\mathbf{z}}_{k+1}^k,\, \mathcal{X}) \right\|_{\mathbf{P}_{k+1}^k}^2 + \sum_{(l,j) \in \mathcal{C}} \left\| r_{\mathcal{C}}(\hat{\mathbf{z}}_l^j,\, \mathcal{X}) \right\|_{\mathbf{P}_l^j}^2 \right\}$$

Linearization

$$\min_{\delta\mathcal{X}} \left\{ \left(\mathbf{b}_p - \mathbf{\Lambda}_p \hat{\mathcal{X}}\right) + \sum_{k \in \mathcal{D}} \left\| r_{\mathcal{D}}(\hat{\mathbf{z}}_{k+1}^k,\, \hat{\mathcal{X}}) + \mathbf{H}_{k+1}^k \delta\mathcal{X} \right\|_{\mathbf{P}_{k+1}^k}^2 + \sum_{(l,j) \in \mathcal{C}} \left\| r_{\mathcal{C}}(\hat{\mathbf{z}}_l^j,\, \hat{\mathcal{X}}) + \mathbf{H}_l^j \delta\mathcal{X} \right\|_{\mathbf{P}_l^j}^2 \right\}$$

Prior IMU residual Reprojection error

Autonomous Quadrotor Flight



Self-Calibrating Multi-Sensor Fusion

Self-Calibrating Multi-Camera Visual-Inertial Fusion for Autonomous MAVs

Zhenfei Yang, Tianbo Liu, and Shaojie Shen

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

High resolution video available at:
http://www.ece.ust.hk/~eeshaojie/iros2016zhenfei.mp4

Summary of Tightly-Coupled Fusion

- Accuracy ☺
- Frequency ☺
- Latency ☺
- ~~Sensor synchronization & timestamp accuracy ☹~~
- Delayed and out-of-order measurements ☺
- Estimator initialization ☺
- Sensor calibration ☺
- Different measurement models with uncertainties ☺
- Robustness to outliers 😐
- Computational efficiency 😐

Conclusion

- Why sensor fusion?
- What methods are available?
  - Filtering-based
  - Optimization-based
  - Loosely-coupled
  - Tightly-coupled
- What are the special design considerations?
  - Multiple measurement models
  - Delayed and out-of-order measurements
  - Sensor calibration
  - Estimator initialization
  - Computational efficiency