Visual-Inertial State Estimation

Guoquan (Paul) Huang

Mechanical Engineering
Electrical and Computer Engineering
Computer and Information Sciences
University of Delaware

1

Motivating Example: AR/VR

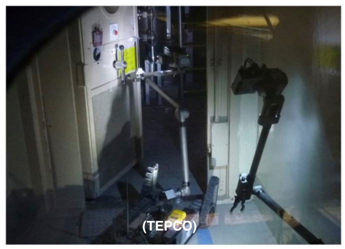
2

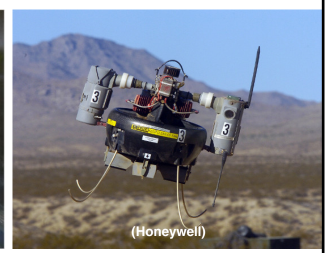
And more…

- Camera is an ideal aiding-source for *inertial navigation* in GPS-denied environments:
  - Light-weight, low-cost, but providing rich info.
- Numerous potential applications:
  - Autonomous driving
  - Emergence response (e.g., in Fukushima)
  - Situational awareness
  - Visually impaired navigation


(Blinput)

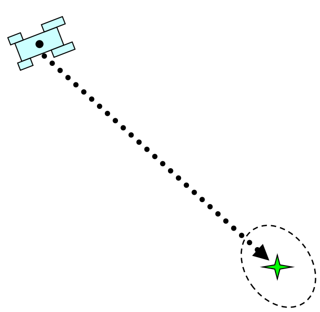

(Waymo)


(TEPCO)


(Honeywell)

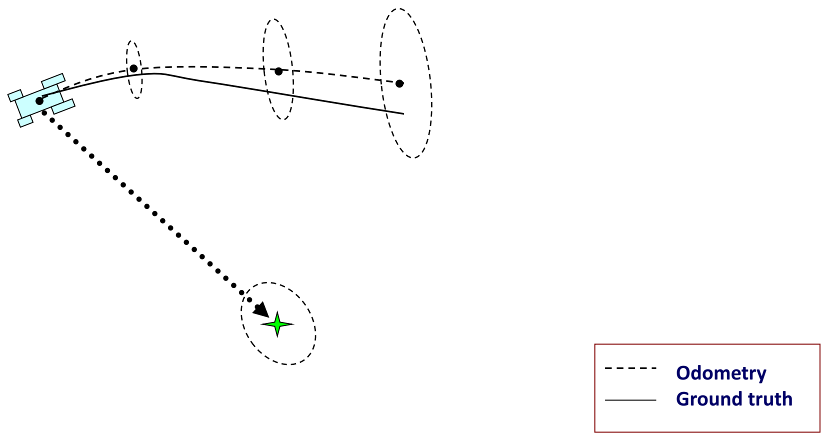
Simultaneous Localization and Mapping (SLAM)

- Detect landmarks (features) in the environment
- Jointly estimate landmark positions and robot pose



Simultaneous Localization and Mapping (SLAM)

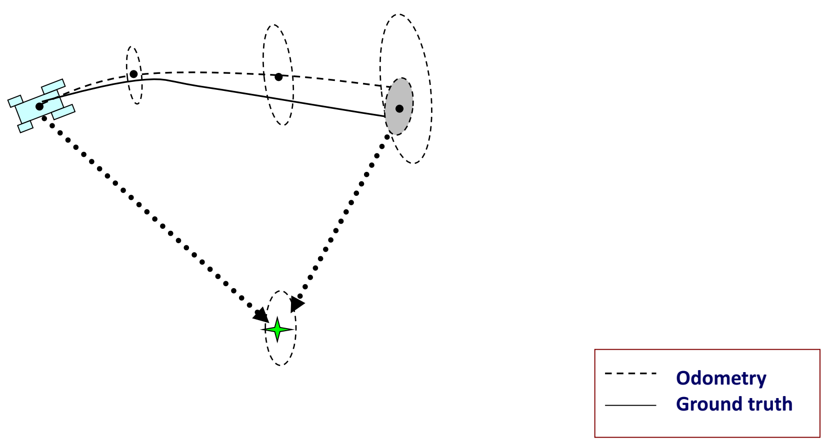
- Detect landmarks (features) in the environment
- Jointly estimate landmark positions and robot pose



5

Simultaneous Localization and Mapping (SLAM)

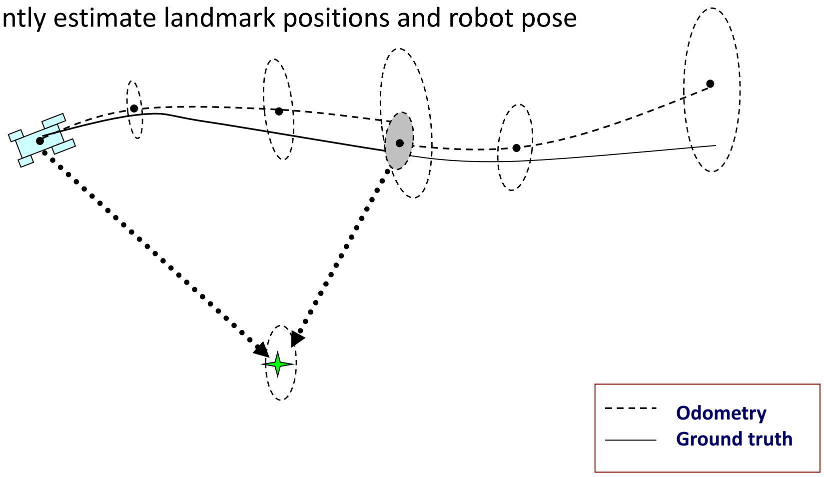
- Detect landmarks (features) in the environment
- Jointly estimate landmark positions and robot pose



6

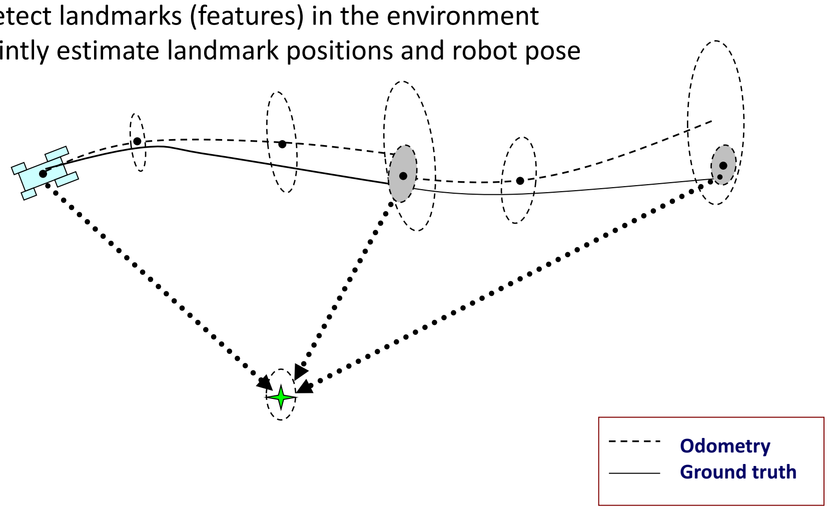
Simultaneous Localization and Mapping (SLAM)

- Detect landmarks (features) in the environment
- Jointly estimate landmark positions and robot pose

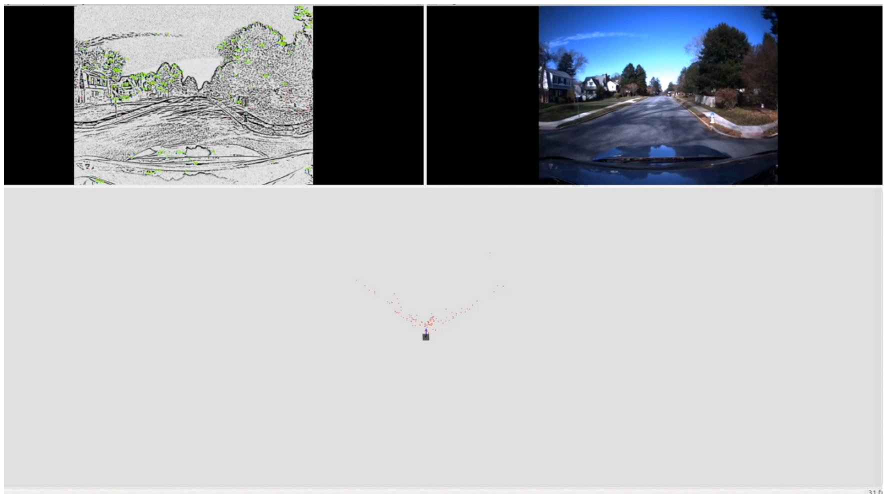


Simultaneous Localization and Mapping (SLAM)

- Detect landmarks (features) in the environment
- Jointly estimate landmark positions and robot pose



SLAM in Action [IROS 18a]



State Estimation

- Nonlinear estimation:
  - Nonlinear system: $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k)$ $\quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$
  - Nonlinear measurement: $\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k$ $\quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$

$$\mathbf{z}_0 \;\; \mathbf{z}_1 \;\; \mathbf{z}_2 \;\; \cdots \;\; \mathbf{z}_{k-N} \;\; \cdots \;\; \mathbf{z}_k$$

$$\longrightarrow t$$

$$\mathbf{x}_0 \;\; \mathbf{x}_1 \;\; \mathbf{x}_2 \;\; \cdots \;\; \mathbf{x}_{k-N} \;\; \cdots \;\; \mathbf{x}_k$$

- Filtering: posterior pdf $p(\mathbf{x}_k | \mathbf{z}_{0:k})$
  - Linearized filters:
    - Extended Kalman Filter (EKF)
    - Unscented Kalman Filter (UKF)
  - Nonparametric method: Particle Filter (PF)

Minimum Mean Square Error (MMSE) Estimator:
$\hat{\mathbf{x}}_{k|k} = \mathbb{E}(\mathbf{x}_k | \mathbf{z}_{0:k})$

- Smoothing:
  - Sliding Window Filter (SWF): $p(\mathbf{x}_{k-N:k} | \mathbf{z}_{0:k})$
  - Batch Maximum A Posteriori (MAP) Estimator: $\hat{\mathbf{x}}_{0:k|k} = \arg\max p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$

SLAM: A State Estimation Problem

- State vector (1 landmark):

$$\mathbf{x}_k = \left[\mathbf{x}_{R_k}^T \;\; \mathbf{p}_L^T\right]^T = \left[\mathbf{p}_{R_k}^T \;\; \phi_k \;\; \mathbf{p}_L^T\right]^T \in \mathcal{R}^{5\times 1}$$

- Motion model:

2x2 rotation matrix

$$\boxed{\begin{aligned} \mathbf{p}_{R_{k+1}} &= \mathbf{p}_{R_k} + \mathbf{C}(\phi_k)\ {}^{R_k}\delta\mathbf{p}_k \\ \phi_{k+1} &= \phi_k + \delta\phi_k \\ \mathbf{p}_{L_{k+1}} &= \mathbf{p}_{L_k} \end{aligned}}$$

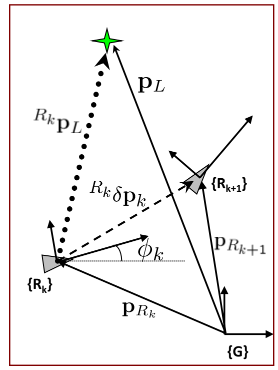
$\mathbf{u}_k = \{{}^{R_k}\delta\mathbf{p}_k, \delta\phi_k\}$ = odometry

$$\Rightarrow \mathbf{x}_{k+1} := \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k + \mathbf{w}_k)$$

- Measurement model:

$$\boxed{\mathbf{z}_k = \mathrm{h}({}^{R_k}\mathbf{p}_L) + \mathbf{v}_k} = \mathrm{h}(\mathbf{x}_k) + \mathbf{v}_k$$

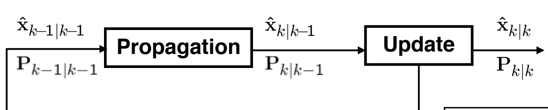
$${}^{R_k}\mathbf{p}_L = \mathbf{C}^T(\phi_k)(\mathbf{p}_L - \mathbf{p}_{R_k})$$



This is a state estimation problem and many different estimators available!

11

Extended Kalman Filter (EKF)



$\hat{\mathbf{x}}_{i|j}, \mathbf{P}_{i|j}$: state and covariance estimates at time *i* given measurements up to time *j*

- EKF Propagation:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1})$$

$$\mathbf{P}_{k|k-1} = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T$$

$$\boxed{\begin{aligned} \mathbf{\Phi}_{k-1} &= \nabla_{\mathbf{x}_{k-1}}\mathbf{f}\Big|_{\{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{0}\}} \\ \mathbf{G}_{k-1} &= \nabla_{\mathbf{w}_{k-1}}\mathbf{f}\Big|_{\{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{0}\}} \end{aligned}}$$

- EKF Update:

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathrm{h}(\hat{\mathbf{x}}_{k|k-1})\right)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T$$

$$\boxed{\mathbf{H}_k = \nabla_{\mathbf{x}_k}\mathbf{h}\Big|_{\hat{\mathbf{x}}_{k|k-1}}}$$

12

Observability Analysis

- Observability:
  - Given inputs/outputs of a dynamical system, to determine initial state.
  - For SLAM system, inputs typically include proprioceptive measurements (e.g., IMU, odometer), and outputs are exteroceptive measurements (e.g., LiDAR, camera).
  - Provides system insights about which state variables are *possible* to estimate and which not.
- Nonlinear observability analysis:
  - Lie derivative-based method [Hermann and Krener 1977]
- Linear observability analysis:
  - Observability matrix formed w/ system and measurement Jacobians:

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_{k_o} \\ \mathbf{H}_{k_o+1}\mathbf{\Phi}_{k_o} \\ \vdots \\ \mathbf{H}_{k_o+m}\mathbf{\Phi}_{k_o+m-1}\cdots\mathbf{\Phi}_{k_o} \end{bmatrix}$$

Observability-Constrained (OC)-EKF SLAM [IJRR 10]

- Sydney Victoria Park:
  - 4-wheeled vehicle with GPS, odometer, and laser
  - 4 km trajectory (**26 minutes**): 6969 robot poses and 151 landmarks
  - Real-time performance: OC-EKFs vs. iSAM (state of the art) [Kaess et al. 08]

Maximum A Posteriori (MAP) Estimator

- Maximize the posterior pdf

$$\mathbf{z}_0 \quad \mathbf{z}_1 \quad \mathbf{z}_2 \quad \cdots \quad \mathbf{z}_{k-N} \quad \cdots \quad \mathbf{z}_k$$

$$t$$

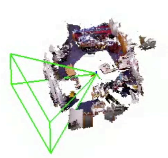
$$\mathbf{x}_0 \quad \mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_{k-N} \quad \cdots \quad \mathbf{x}_k$$

$$\hat{\mathbf{x}}_{0:K|0:K} = \arg\max p\left(\mathbf{x}_{0:K}|\mathbf{z}_{0:K}\right)$$

$$\hat{\mathbf{x}}_{0:K|0:K} = \arg\min \left(\sum_{k=0}^{K} \sum_{m=1}^{M_k} \frac{1}{2} ||\mathbf{z}_{km} - h\left(\mathbf{x}_k\right)||^2_{\mathbf{R}_{km}} + \sum_{k=0}^{K-1} \frac{1}{2} ||\mathbf{x}_{k+1} - \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)||^2_{\mathbf{Q}_k} + \frac{1}{2}||\mathbf{x}_0 - \mathbf{x}_{init}||^2_{\mathbf{P}_0} \right)$$

15

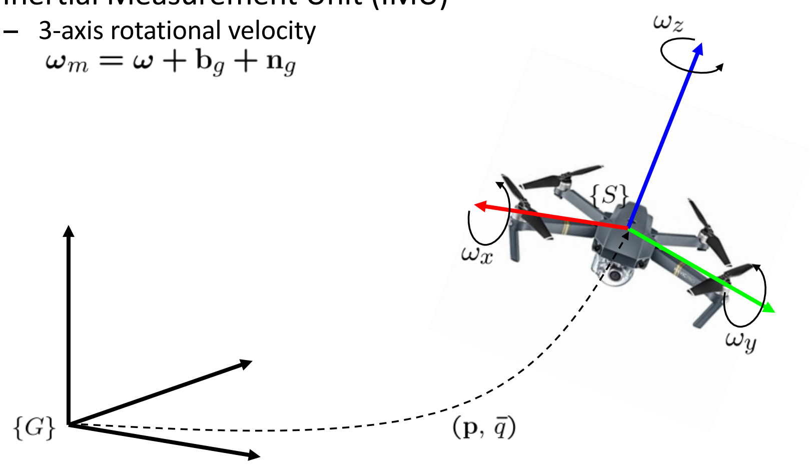
Graph SLAM [ECMR 13a]



16

Visual-Inertial Navigation System (VINS)

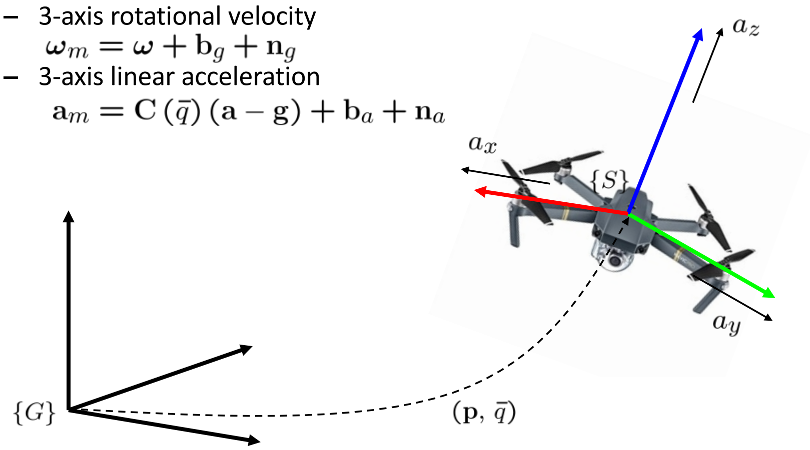
- Estimate 6 DOF pose: position $\mathbf{p}$ & orientation $\bar{q}$
- Inertial Measurement Unit (IMU)
  - 3-axis rotational velocity
    $\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g$



17

Visual-Inertial Navigation System (VINS)

- Estimate 6 DOF pose: position $\mathbf{p}$ & orientation $\bar{q}$
- Inertial Measurement Unit (IMU)
  - 3-axis rotational velocity
    $\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g$
  - 3-axis linear acceleration
    $\mathbf{a}_m = \mathbf{C}(\bar{q})(\mathbf{a} - \mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a$

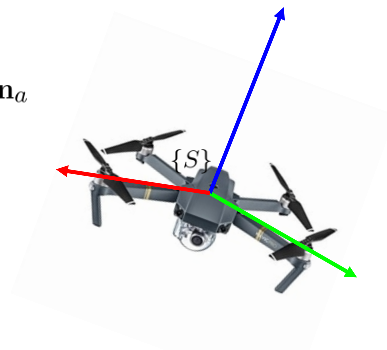


18

Visual-Inertial Navigation System (VINS)

- Estimate 6 DOF pose: position $\mathbf{p}$ & orientation $\bar{q}$
- Inertial Measurement Unit (IMU)
  - 3-axis rotational velocity
    $\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g$
  - 3-axis linear acceleration
    $\mathbf{a}_m = \mathbf{C}\left(\bar{q}\right)\left(\mathbf{a} - \mathbf{g}\right) + \mathbf{b}_a + \mathbf{n}_a$
- Kinematic model of motion

$$\begin{aligned}
\dot{\bar{q}} &= \frac{1}{2}\Xi\left(\bar{q}\right)\boldsymbol{\omega} \\
\dot{\mathbf{p}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \mathbf{a} \\
\dot{\mathbf{b}} &= \mathbf{w}, \quad \mathbf{w} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}\right)
\end{aligned}$$



However, integration of noise & bias causes ***large drift*** *in the pose estimate!*

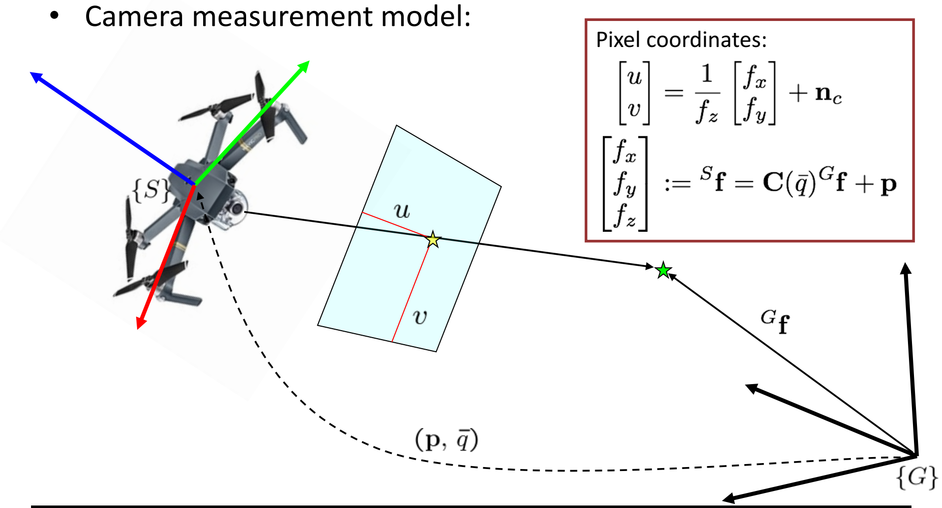
Visual-Inertial Navigation System (VINS)

- Use IMU & camera to estimate $\bar{q}$, $\mathbf{p}$, $\mathbf{v}$, $\mathbf{b}$
- Camera measurement model:

Pixel coordinates:

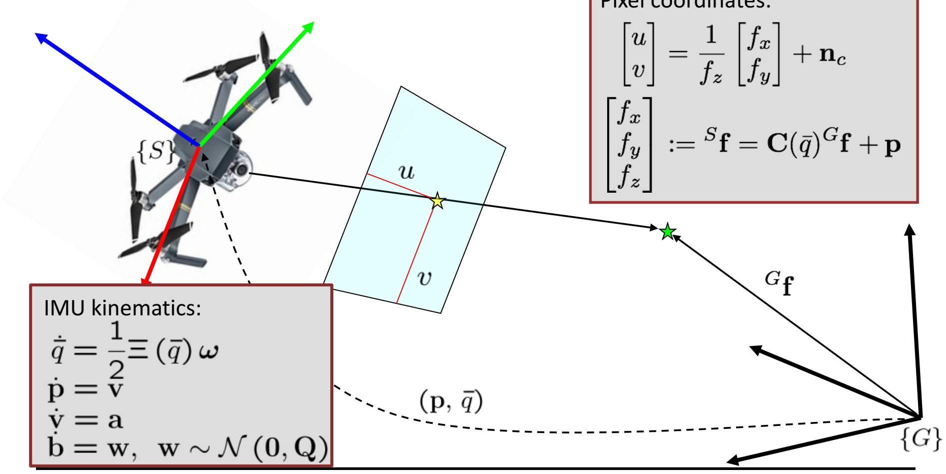
$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{f_z}\begin{bmatrix} f_x \\ f_y \end{bmatrix} + \mathbf{n}_c$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} := {}^{S}\mathbf{f} = \mathbf{C}(\bar{q})\,{}^{G}\mathbf{f} + \mathbf{p}$$



Visual-Inertial Navigation System (VINS)

- Use IMU & camera to estimate $\bar{q},\ \mathbf{p},\ \mathbf{v},\ \mathbf{b}$



Pixel coordinates:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{f_z}\begin{bmatrix} f_x \\ f_y \end{bmatrix} + \mathbf{n}_c$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} := {}^{S}\mathbf{f} = \mathbf{C}(\bar{q})\,{}^{G}\mathbf{f} + \mathbf{p}$$

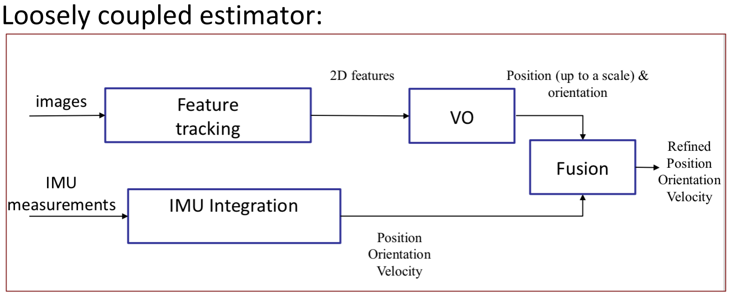
IMU kinematics:

$$\dot{\bar{q}} = \frac{1}{2}\Xi(\bar{q})\,\boldsymbol{\omega}$$
$$\dot{\mathbf{p}} = \mathbf{v}$$
$$\dot{\mathbf{v}} = \mathbf{a}$$
$$\dot{\mathbf{b}} = \mathbf{w},\quad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$

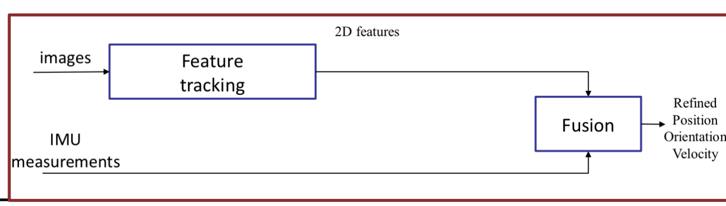
21

Loosely vs. Tightly Coupled Estimation

- Loosely coupled estimator:



- Tightly coupled estimator:



22

Filtering: MSCKF [Mourikis 2007]

- Key idea of Multi-State Constraints Kalman Filter (MSCKF):
  - Estimate a sliding window of IMU states without keeping features in the state vector, while tracking visual features in the window to provide motion constraints that are used for EKF update.
- State vector: $\mathbf{x} = \left[{}^{I}_{G}\bar{q}^{\top}, \mathbf{b}_g{}^{\top}, {}^{G}\mathbf{v}_I{}^{\top}, \mathbf{b}_a{}^{\top}, {}^{G}\mathbf{p}_I{}^{\top}, \underbrace{\mathbf{x}_c{}^{\top}}_{\text{Cloned state of past poses}}\right]^{\top}$
- IMU propagation:

$$\left.\begin{array}{lrcl} \text{Attitude} & {}^{I}\dot{\bar{q}}_G(t) & = & \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t))\,{}^{I}\bar{q}_G(t) \\ \text{Velocity-Position} & {}^{G}\dot{\mathbf{p}}_I(t) & = & {}^{G}\mathbf{v}_I(t) \quad , \quad {}^{G}\dot{\mathbf{v}}_I(t) = {}^{G}\mathbf{a}(t) \\ \text{Gyroscope} & \boldsymbol{\omega}_m & = & \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g \\ \text{Accelerometer} & \mathbf{a}_m & = & \mathbf{C}({}^{I}\bar{q}_G)({}^{G}\mathbf{a} - {}^{G}\mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a \\ \text{Biases} & \dot{\mathbf{b}}_g(t) & = & \mathbf{n}_{wg}(t) \quad , \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \end{array}\right\} \Rightarrow$$

$$\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}}) \qquad \dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{Q} \qquad \mathbf{F} = \nabla_{\mathbf{x}} f$$

MSCKF Update

- Observation of feature *j* from camera pose *i*:

$$\mathbf{z}_i^{(j)} = \frac{1}{{}^{C_i}Z_j}\begin{bmatrix}{}^{C_i}X_j \\ {}^{C_i}Y_j\end{bmatrix} + \underbrace{\mathbf{n}_i^{(j)}}_{\text{Measurement noise}} \qquad {}^{C_i}\mathbf{p}_j = \begin{bmatrix}{}^{C_i}X_j \\ {}^{C_i}Y_j \\ {}^{C_i}Z_j\end{bmatrix} = \underbrace{{}^{C_i}_{G}\mathbf{C}}_{\text{Rotation between world and camera frames}}(\underbrace{{}^{G}\mathbf{p}_{\ell_j}}_{\text{Feature position}} - \underbrace{{}^{G}\mathbf{p}_{C_i}}_{\text{Camera position}})$$

- Measurement residual:

$$\mathbf{r}_i^{(j)} \simeq \mathbf{H}_{\mathsf{OF}_i}^{(j)}\widetilde{\mathbf{X}} + \mathbf{H}_{f_i}^{(j)}\,{}^{G}\widetilde{\mathbf{p}}_{\ell_j} + \mathbf{n}_i^{(j)} \Rightarrow$$
$$\mathbf{r}^{(j)} \simeq \underbrace{\mathbf{H}_{\mathsf{OF}}^{(j)}\widetilde{\mathbf{X}}}_{\text{State Error}} + \mathbf{H}_{f}^{(j)}\,\underbrace{{}^{G}\widetilde{\mathbf{p}}_{\ell_j}}_{\text{Feature position error}} + \mathbf{n}^{(j)}$$

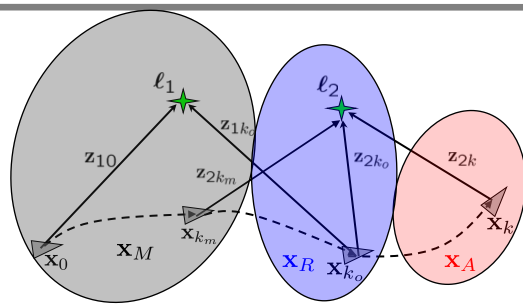
(stack all residuals of feature *j*)

  - Project onto **left nullspace** of $\mathbf{H}_f^{(j)}$ to remove effect of feature position error

$$\boxed{\mathbf{r}_o^{(j)} = \mathbf{U}^T\mathbf{r}^{(j)} = \mathbf{H}_o^{(j)}\widetilde{\mathbf{X}} + \mathbf{n}_o^{(j)}}$$

  - Perform EKF update
- Used in Google ARCore (and will be in next Oculus devices)

Smoothing: Sliding Window BA



- Formulate as a MAP (or nonlinear least squares) optimization:

$$\min_{\{\mathbf{x}_i,\ell_j\}} \underbrace{\sum_{i,j} ||\mathbf{z}_{ij} - h(\mathbf{x}_i,\ell_j)||^2_{\mathbf{R}_{ij}}}_{\text{Reprojection residuals}} + \underbrace{\sum_k ||\mathbf{x}_k - f(\mathbf{x}_{k-1},\mathbf{u}_m)||^2_{\mathbf{Q}_k}}_{\text{IMU residuals}}$$

  - Visual tracking [Lucas 1981], IMU preintegration [IJRR 19a], marginalization [IROS 11]
  - Iterative solvers such as G-N and L-M
  - Examples: OKVIS [Leutenegger et al. 2014], VINS-Mono [Qin et al. 2018]

VINS Review [ICRA 19]

2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Visual-Inertial Navigation: A Concise Review

Guoquan Huang

***Abstract*— As inertial and visual sensors are becoming ubiquitous, visual-inertial navigation systems (VINS) have prevailed in a wide range of applications from mobile augmented reality to aerial navigation to autonomous driving, in part because of the complementary sensing capabilities and the decreasing costs and size of the sensors. In this paper, we survey thoroughly the research efforts taken in this field and strive to provide a concise but complete review of the related work – which is unfortunately missing in the literature while being greatly demanded by researchers and engineers – in the hope to accelerate the VINS research and beyond in our society as a whole.**

I. INTRODUCTION

Over the years, inertial navigation systems (INS) [1, 2] have been widely used for estimating the 6DOF poses (positions and orientations) of sensing platforms (e.g., autonomous vehicles), in particular, in GPS-denied environments such as underwater, indoor, in the urban canyon, and on other planets. Most INS rely on a 6-axis inertial measurement unit (IMU) that measures the local linear acceleration and angular velocity of the platform to which it is rigidly connected. With the recent advancements of hardware design and manufacturing, low-cost light-weight micro-electro-mechanical (MEMS) IMUs have become ubiq-

filter (UKF) [23–25], the batch or incremental smoother [26, 27], and (window) optimization-based approaches [28–31]. Among these, the EKF-based methods remain popular because of its efficiency. For example, as a state-of-the-art solution of VINS on mobile devices, Project Tango [32] (or ARCore [12]) appears to use an EKF to fuse the visual and inertial measurements for motion tracking. Nevertheless, recent advances of preintegration have also allowed for efficient inclusion of high-rate IMU measurements in graph optimization-based formulations [29, 30, 33–35].

As evident, VINS technologies are emerging, largely due to the demanding mobile perception/navigation applications, which has given rise to a rich body of literature in this area. However, to the best of out knowledge, there is *no* contemporary literature review of VINS, although there are recent surveys broadly about SLAM [16, 36] while not specializing on VINS. This has made difficult for researchers and engineers in both academia and industry, to effectively find and understand the most important related work to their interests, which we have experienced over the years when we are working on this problem. For this reason, we are striving to bridge this gap by: (i) offering a concise (due

VINS Workshop @ IROS 2019

Visual-Inertial Navigation: Challenges and Applications

IROS 2019 Full-day Workshop: November 8, 2019, Macau, China

Updates

- (7/3) We are pleased to announce that LORD Sensing will sponsor the **LORD Best Paper Award** (with MicroStrain IMUs as prize)!
- (7/2) Please prepare papers following the IROS template available in LaTeX and MS Word.
- (7/1) We are pleased to announce that there will be a **Special Issue in the IET Cyber-systems and Robotics**, which will invite some of best papers presented at this workshop.

Overview

As cameras and IMUs are becoming ubiquitous, visual-inertial navigation systems (VINS
estimation, hold great potentials in a wide range of applications from augmented reality (
driving, in part because of the complementary sensing capabilities and the decreasing cos
inertial navigation, alongside with SLAM, has witnessed tremendous progress in the past
of visual-inertial systems remain poorly explored, greatly hindering the widespread depl
example, many VINS algorithms are yet not robust to high dynamics and poor lighting c
long-term, large-scale operations, in particular, in life-critical scenarios; and yet they are u
understandings to support high-level decision making. This workshop brings together re
from both academia and industry, to share their insights and thoughts on the R&D of VIN
forward the latest breakthroughs and cutting-edge research on visual-inertial navigation
technical challenges and future research directions for the community, and to identify nev

Call for Contributions

We welcome submissions of papers describing VINS-related work in progress, prelimina
experiences. All submitted papers will be reviewed by at least two experts (see Program (

Organizers

- **Guoquan (Paul) Huang**, University of Delaware
- **Shaojie Shen**, Hong Kong University of Science and Technology
- **Michael Kaess**, CMU
- **Stergios Roumeliotis**, University of Minnesota
- **John Leonard**, MIT

Invited Speakers and Panelists (confirmed)

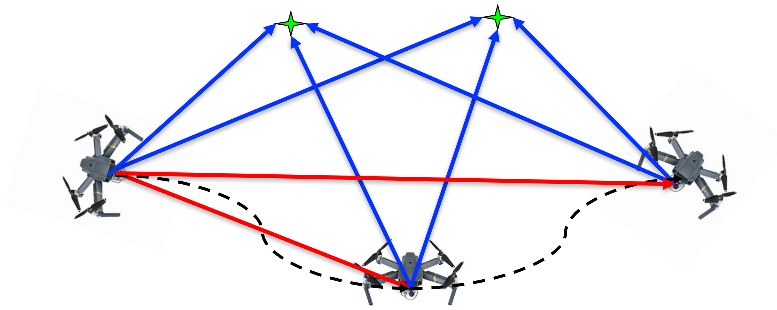
- **Luca Carlone**, MIT
- **Jakob Engel**, Facebook Oculus
- **Maurice Fallon**, Oxford
- **Jonathan Kelly**, University of Toronto
- **Giuseppe Loianno**, NYU
- **Laurent Kneip**, Shanghai Tech
- **Davide Scaramuzza**, University of Zurich
- **Guofeng Zhang**, Zhejiang University
- **Ross Hartley**, Amazon (University of Michigan)

27

Visual-Inertial Estimation Research @ My Lab (RPNG)

28

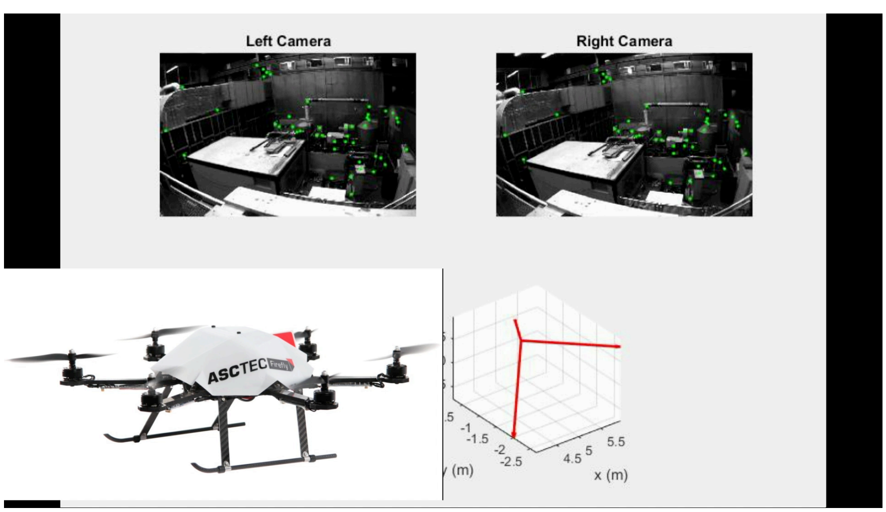
Optimal-State-Constraint (OSC)-EKF [ISRR 15; ARL 17]



- Key ideas:
  - Process a sliding window of images *only* to infer **optimal** state constraints btw. corresponding camera poses
  - Perform EKF propagation using IMU measurements, and EKF update using the inferred state constraints
  - As a result, *no* need to store features in the state vector, yielding *constant* complexity

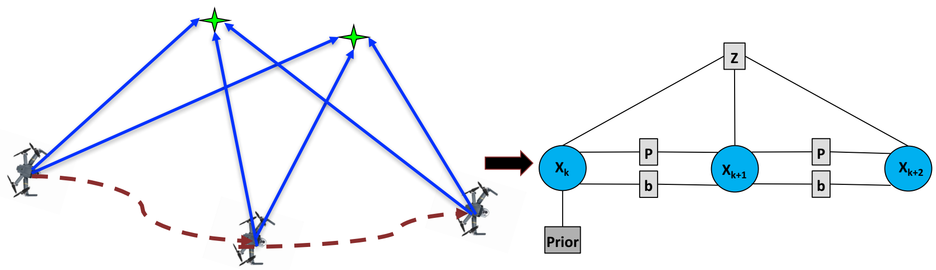
29

Results: OSC-EKF VIO



30

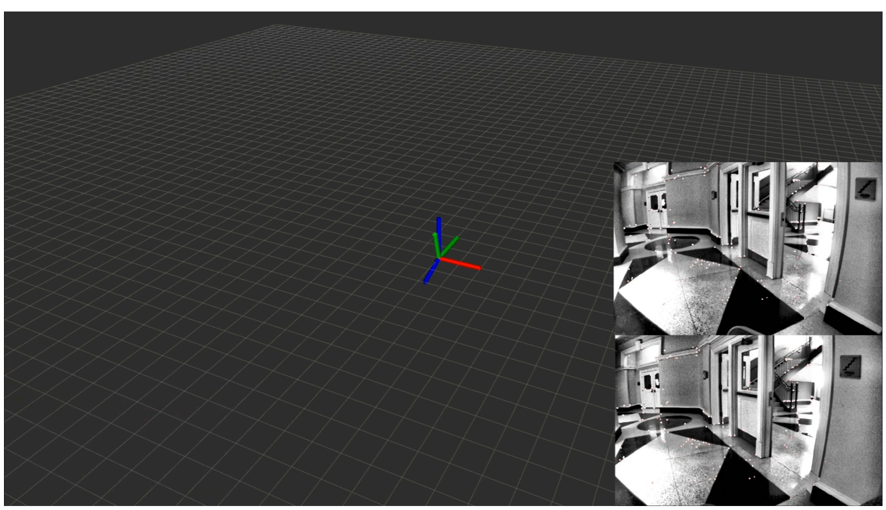
Closed-form Preintegration (CPI) for Graph-VINS [WAFR 16; IJRR 19a]



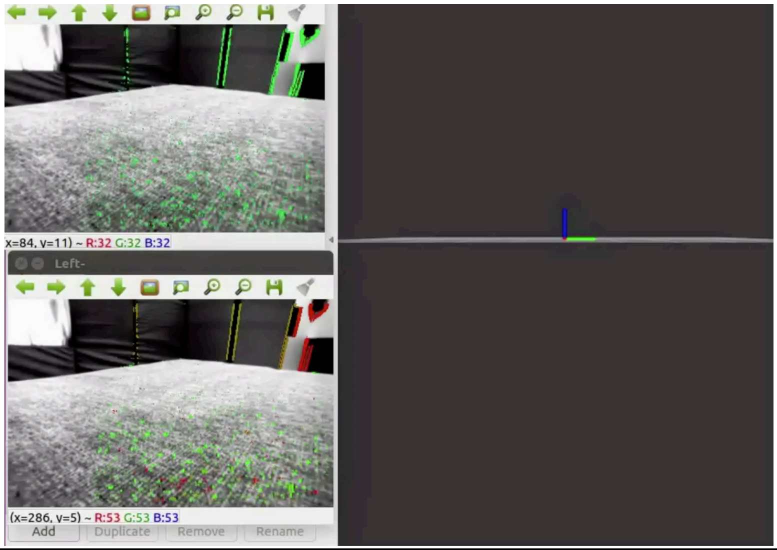
- Sliding-window BA to *tightly* fuse visual and inertial measurements
  - Analytical continuous-time IMU preintegration: (i) piecewise constant measurement, and (ii) piecewise constant local acceleration
  - Perform bundle adjustment (BA) to optimally estimate a sliding window of states and features detected
  - Marginalize out features to bound computational complexity

Open source: https://github.com/rpng/cpi

Results: Graph-VINS [IJRR 19a]

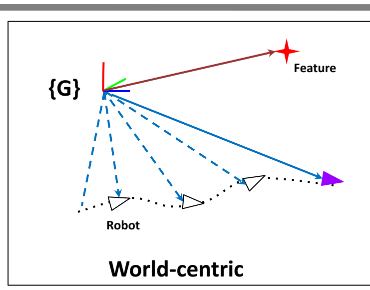


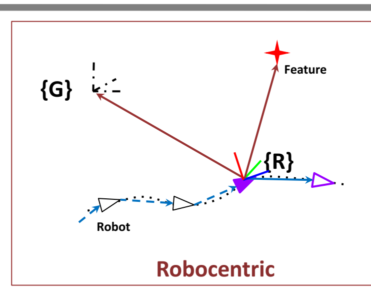
Direct VINS [ICRA 17a, IJRR19a]



33

Robocentric VIO [IROS 18a, IJRR19b]





- Robocentric VIO (R-VIO) within the MSCKF framework:
  - State includes local gravity and a sliding window of relative poses
  - Close-form IMU preintegration (for propagation)
  - Inverse depth-based measurement model (for update)
  - Composition is employed to shift robocentric frame after update

Open source: https://github.com/rpng/r-vio

34

Results: R-VIO [IJRR 19b]

Robocentric
Visual-Inertial
Navigation

Results: R-VIO [IJRR 19b]

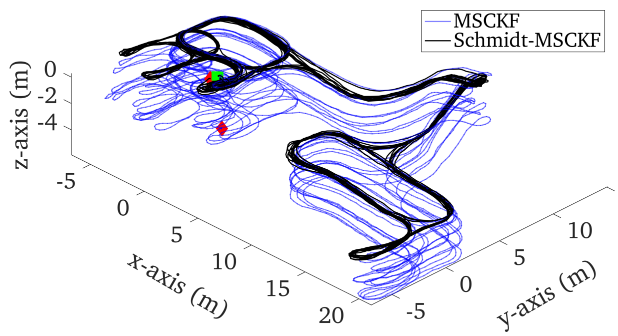
Robocentric Visual-Inertial Odometry

Zheng Huai and Guoquan Huang

Robot Perception and Navigation Group (RPNG)
University of Delaware

Schmidt MSCKF for VINS w/ Loop Closures [ICRA 19c]

- A novel linear-complexity algorithm for VINS with loop closures:
  - Exploit Schmidt-KF for real-time consistent inclusion of old keyframes by only updating their cross-correlations
  - Leverage MSCKF nullspace-based marginalization, allowing for efficient processing measurements of keyframe-based loop-closures



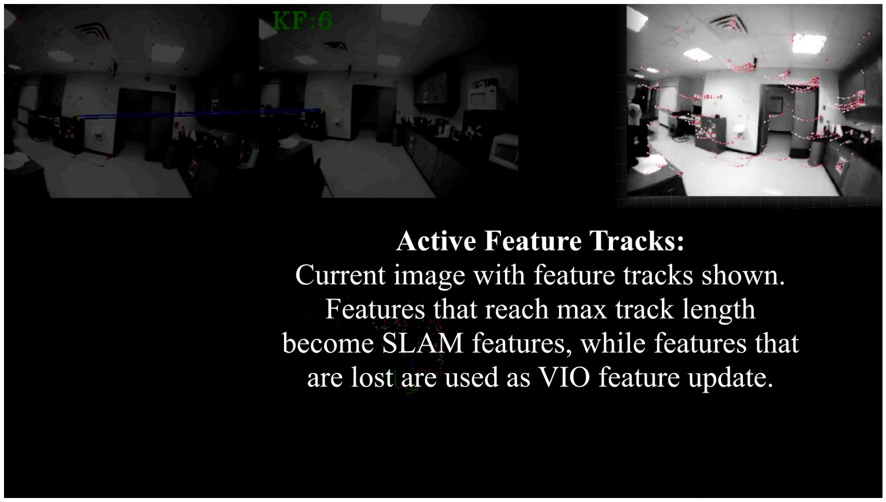
Results: Schmidt MSCKF [ICRA 19c]

A Linear-Complexity EKF for Visual-Inertial Navigation with Loop Closures

Patrick Geneva, Kevin Eckenhoff, and Guoquan Huang

RPNG, University of Delaware, USA

Schmidt-EKF VI-SLAM [CVPR 19]



39

Multi-Camera VINS [ICRA 19a]

Multi-Camera Visual-Inertial Navigation with Online Intrinsic and Extrinsic Calibration

Kevin Eckenhoff, Patrick Geneva,
Jesse Bloecker, and Guoquan Huang

RPNG, University of Delaware, USA

40

Multi-IMU VINS [ICRA 19b]

Sensor-Failure-Resilient Multi-IMU Visual-Inertial Navigation

Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang

RPNG, University of Delaware, USA

41

Observability Analysis and Representations [ICRA 19f; TRO 19]

Model #	Point	Error states	Line	Error states	Plane	Error states
1: General Form	f_1, f_2, f_3, f_4	not minimal	$\mathbf{n}_l, \mathbf{v}_l$	$\delta\boldsymbol{\theta}_l, \delta\phi_l$	$\pi_1, \pi_2, \pi_3, \pi_4$	not minimal
2: Geometric Form	$\mathbf{b_f}, r_\mathbf{f}$	not minimal	$\mathbf{n}_e = \frac{\mathbf{n}_l}{\Vert\mathbf{n}_l\Vert}$ $\mathbf{v}_e = \frac{\mathbf{v}_l}{\Vert\mathbf{v}_l\Vert}$ $d_l = \frac{\Vert\mathbf{n}_l\Vert}{\Vert\mathbf{v}_l\Vert}$	not minimal	$\mathbf{n}_\pi, d_\pi$	not minimal
3: Spherical Form	$\theta_\mathbf{f}, \phi_\mathbf{f}, r_\mathbf{f}$	$\tilde{\theta}_\mathbf{f}, \tilde{\phi}_\mathbf{f}, \tilde{r}_\mathbf{f}$	$\theta_l, \phi_l, \alpha_l, d_l$	$\tilde{\theta}_l, \tilde{\phi}_l, \tilde{\alpha}_l, \tilde{d}_l$	$\theta_\pi, \phi_\pi, d_\pi$	$\tilde{\theta}_\pi, \tilde{\phi}_\pi, \tilde{d}_\pi$
4: Inverse Depth	$\theta_\mathbf{f}, \phi_\mathbf{f}, \lambda_\mathbf{f} = \frac{1}{r_\mathbf{f}}$	$\tilde{\theta}_\mathbf{f}, \tilde{\phi}_\mathbf{f}, \tilde{\lambda}_\mathbf{f}$	$\theta_l, \phi_l, \alpha_l, \lambda_l = \frac{1}{d_l}$	$\tilde{\theta}_l, \tilde{\phi}_l, \tilde{\alpha}_l, \tilde{\lambda}_l$	$\theta_\pi, \phi_\pi, \lambda_\pi = \frac{1}{d_\pi}$	$\tilde{\theta}_\pi, \tilde{\phi}_\pi, \tilde{\lambda}_\pi$
5: *Quaternion*	$\bar{q}_\mathbf{f} = \frac{1}{\sqrt{1+r_\mathbf{f}^2}}\begin{bmatrix}\mathbf{b_f}\\ r_\mathbf{f}\end{bmatrix}$	$\delta\boldsymbol{\theta}_\mathbf{f}$	$\bar{q}_l, d_l$	$\delta\boldsymbol{\theta}_l, \tilde{d}_l$	$\bar{q}_\pi = \frac{1}{\sqrt{1+d_\pi^2}}\begin{bmatrix}\mathbf{n}_\pi\\ d_\pi\end{bmatrix}$	$\delta\boldsymbol{\theta}_\pi$
6: *Closest Point*	$\mathbf{p_f} = r_\mathbf{f}\mathbf{b_f}$	$\mathbf{p_f} = \hat{\mathbf{p}}_\mathbf{f} + \tilde{\mathbf{p}}_\mathbf{f}$	$\mathbf{p}_l = d_l\bar{q}_l$	$\mathbf{p}_l = \hat{\mathbf{p}}_l + \tilde{\mathbf{p}}_l$	$\mathbf{p}_\pi = d_\pi\mathbf{n}_\pi$	$\mathbf{p}_\pi = \hat{\mathbf{p}}_\pi + \tilde{\mathbf{p}}_\pi$

- Unified representations for **points, lines and planes**: (i) quaternion, and (ii) closest point
- Aided INS with combination of all the geometrical features has 4 unobservable directions: global position and global yaw

Features	Unobservable Directions
Single or multiple points	4
Non-parallel lines	
Planes with non-parallel intersections	
Point and line	
Point and plane	
Single line non-parallel to planes	
Plane intersections non-parallel to lines	
Point, line and plane	
Single line	5
Single line parallel to single plane	
Two non-parallel planes	
Single plane	7

Observability Analysis and Representations [ICRA 19f; TRO 19]

Aided Inertial Navigation: Unified Feature Representations and Observability Analysis

Yulin Yang and Guoquan Huang

RPNG, University of Delaware, USA

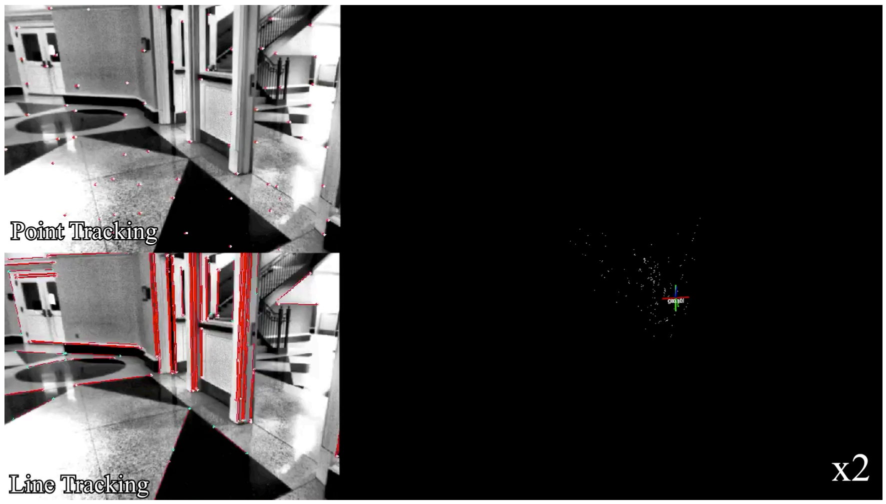
VINS w/ Point and Plane Features [ICRA 19e]

Tightly-Coupled Aided Inertial Navigation with Point and Plane Features

Yulin Yang, Patrick Geneva, Xingxing Zuo,*
Kevin Eckenhoff, Yong Liu* and Guoquan Huang

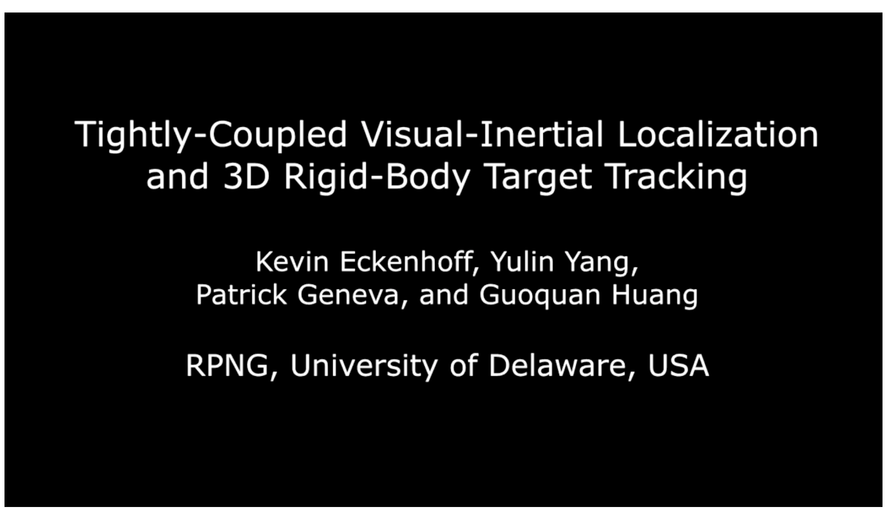
RPNG, University of Delaware, USA
*Institute of Cyber-Systems & Control, Zhejiang University, China

VIO w/ Point and Line Features [IROS 19b]

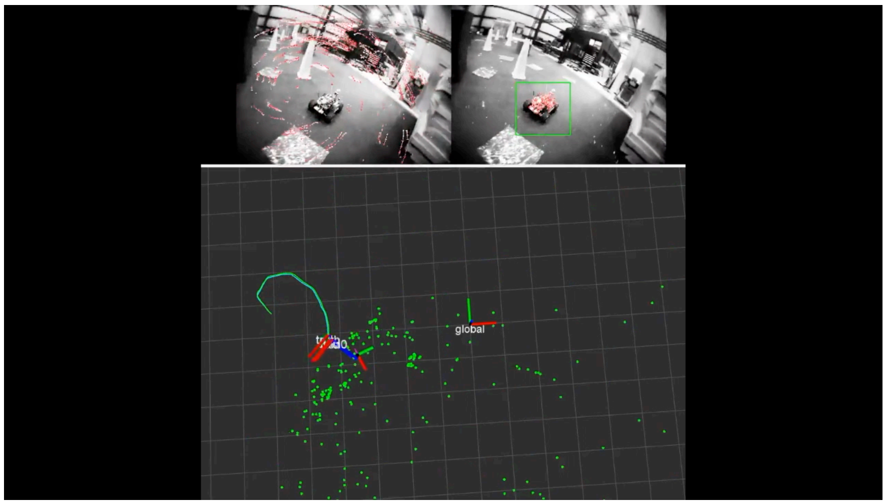


45

Visual-Inertial Localization & Moving Object Tracking [RAL 19a]



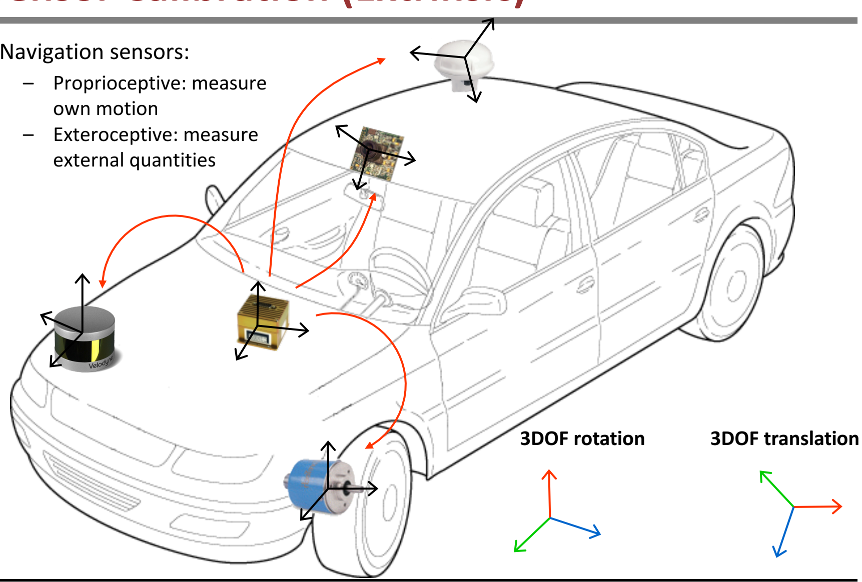
46



47

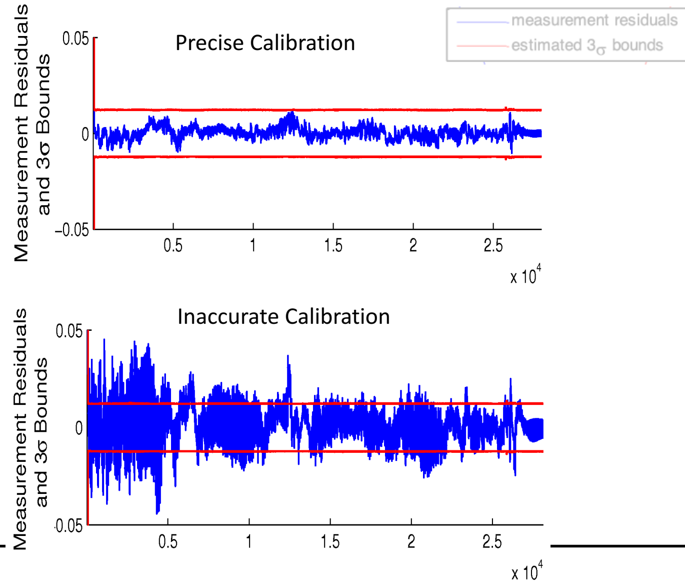
Sensor Calibration (Extrinsic)

- Navigation sensors:
  - Proprioceptive: measure own motion
  - Exteroceptive: measure external quantities



48

Significance of Sensor Calibration



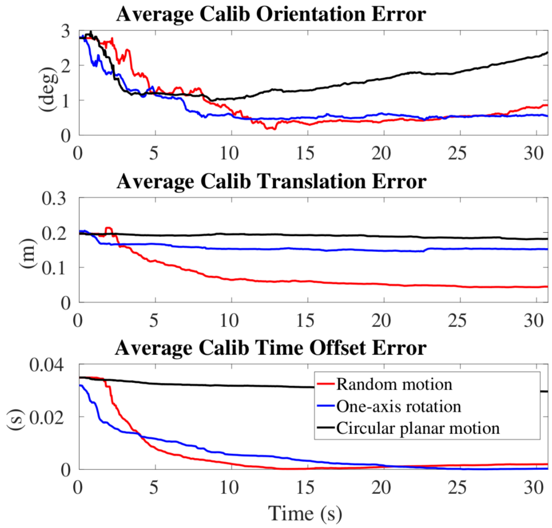
Degenerate Motions for Sensor Calibration [RAL 19b]

Degenerate Motion Analysis for Aided INS with Online Spatial and Temporal Sensor Calibration

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

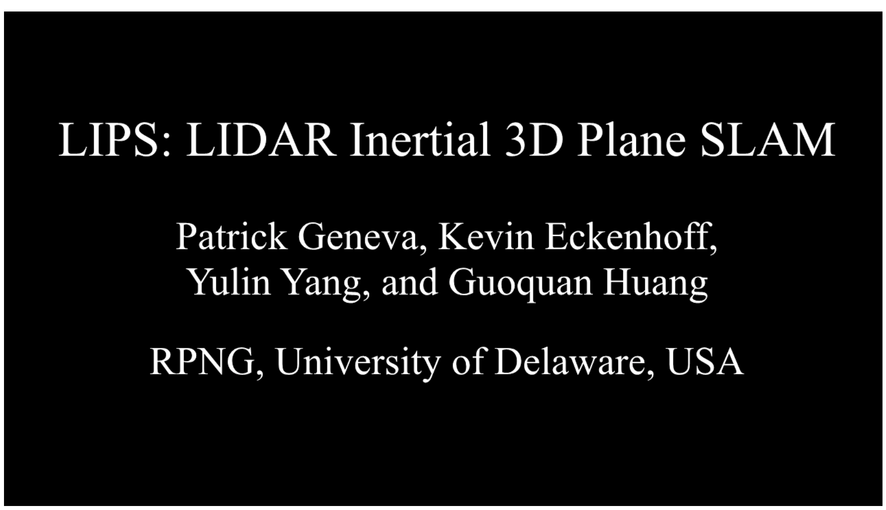
RPNG, University of Delaware, USA

Degenerate Motions for Sensor Calibration [RAL 19b]



51

LiDAR-Inertial Plane SLAM [IROS 18b]



Open source: https://github.com/rpng/lips

52

VINS w/ Prior LiDAR Map [RAL 19c]

Visual-Inertial Localization with
Prior LiDAR Map Constraints

Xingxing Zuo*, Patrick Geneva, Yulin Yang,
Wenlong Ye*, Yong Liu* and Guoquan Huang

RPNG, University of Delaware, USA
*Institute of Cyber-Systems & Control, Zhejiang University, China

53

LiDAR-Inertial-Visual Odometry [IROS 19c]

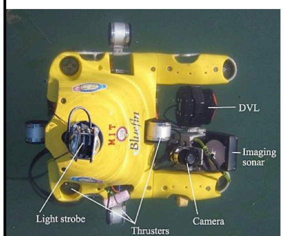
LIC-Fusion: LiDAR-Inertial-Camera Odometry

Xingxing Zuo*, Patrick Geneva, Woosik Lee, Yong Liu*, Guoquan Huang

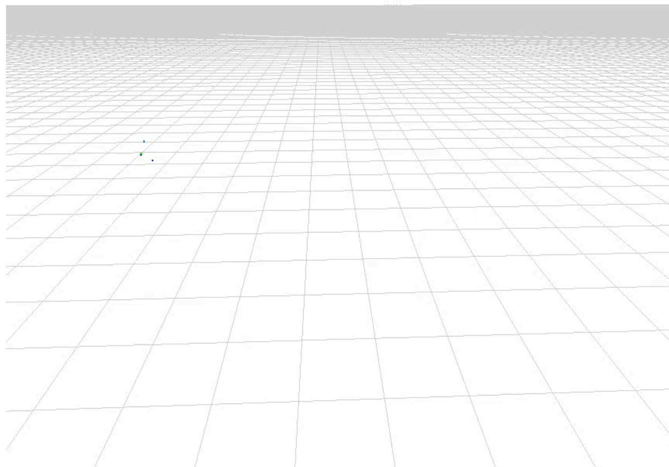
RPNG, University of Delaware, USA
*Institute of Cyber System and Control, Zhejiang University, China

54

Underwater SLAM



Bluefin HAUV

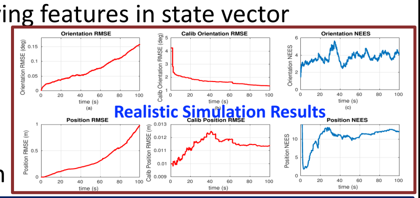


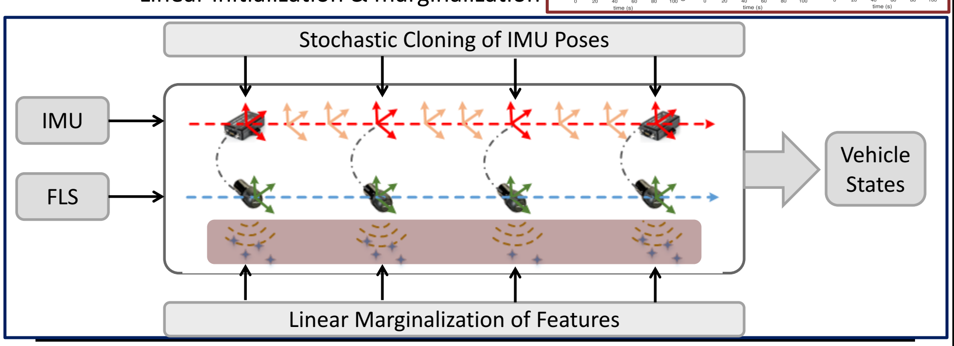
[Joint work with Kaess & Leonard]

55

Acoustic-Inertial Navigation System (AINS) [ICRA 17b]

- **MSCKF-based acoustic-inertial odometry:**
  - Utilize all measurement info w/o having features in state vector
- **Key ideas:**
  - Stochastic cloning
  - EKF update of motion-only meas.
  - Online calibration
  - Linear initialization & marginalization



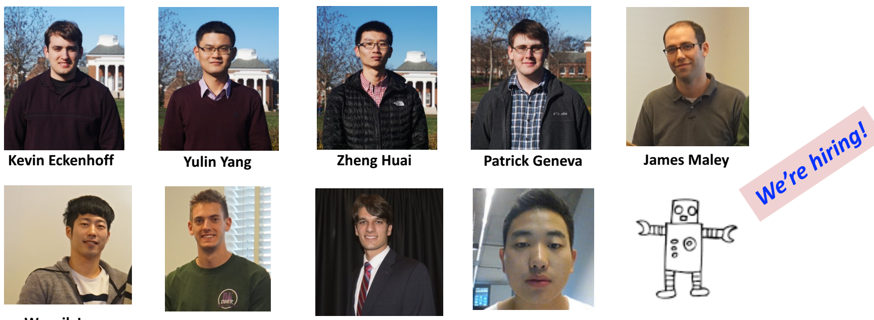


[Johannsson @ Teledyne shared BlueView MB2250 data]

56

Summary

- Visual-inertial estimation is an enabling technology for mobile perception:
  - Current research: state estimation for SLAM/VINS
  - Future research: distributed estimation and perception
- My lab: Robot Perception and Navigation Group (RPNG)
  http://sites.udel.edu/robot https://github.com/rpng



Kevin Eckenhoff, Yulin Yang, Zheng Huai, Patrick Geneva, James Maley

Woosik Lee, Jesse Bloecker, Nate Merrill, Xingxing Zuo (ZJU), Chuchu Chen

- Funded by NSF, DTRA, ARL, NASA, Huawei, Google, Bosch, JRE/MIT, etc.

Thank you!

57