University of Zurich[UZH]

ROBOTICS & PERCEPTION GROUP

Vision Algorithms for Mobile Robotics

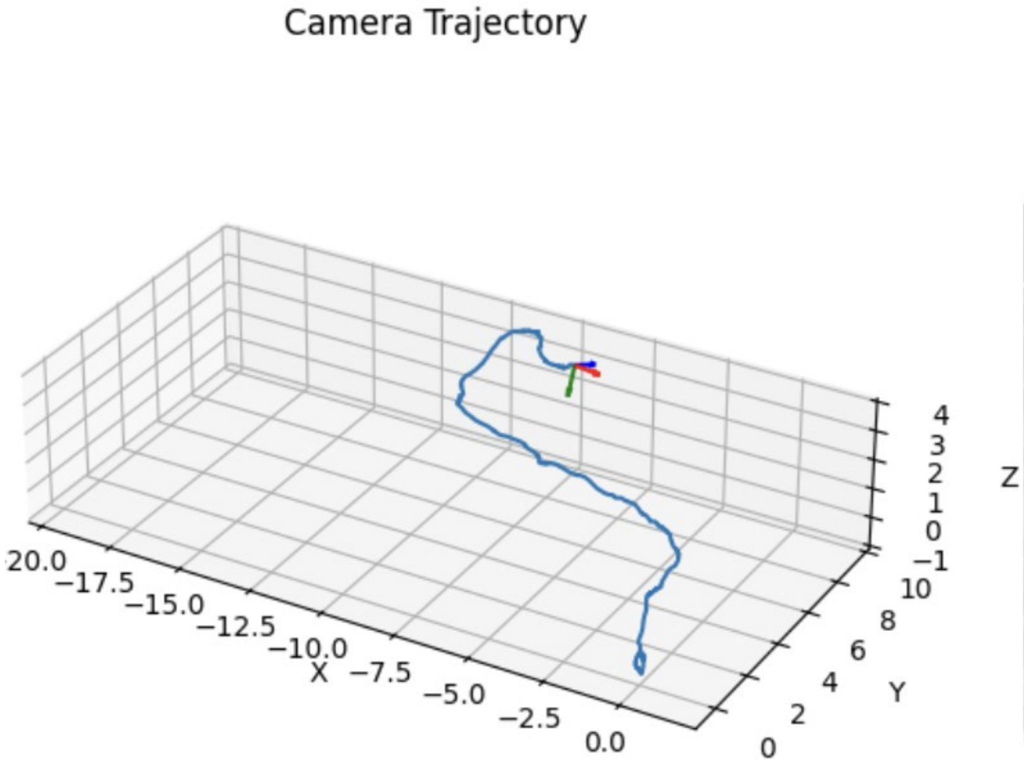
Lecture 13
Visual Inertial Fusion

Davide Scaramuzza

https://rpg.ifi.uzh.ch

Today: Lab Exercise

Visual-inertial fusion



Next week after exercise:
visit of the Robotics and Perception Group

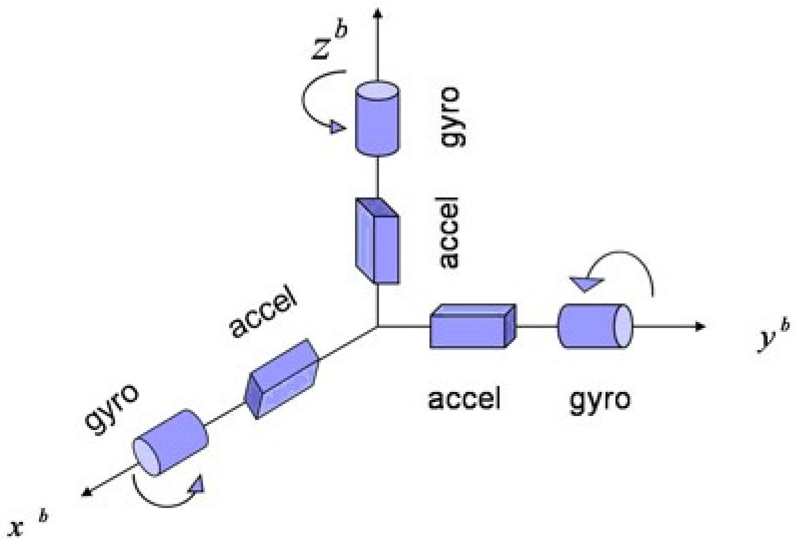
- **Address**: Andreasstrasse 15, 2nd floor, next to **Zurich Oerlikon** train station
- **Webpage**: http://rpg.ifi.uzh.ch – **Limited to registered people.**

Outline

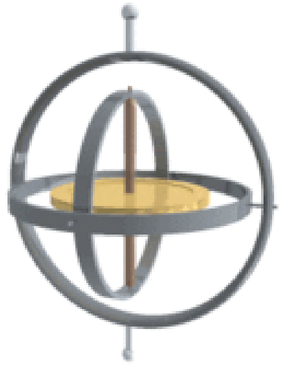
- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

What is an IMU?

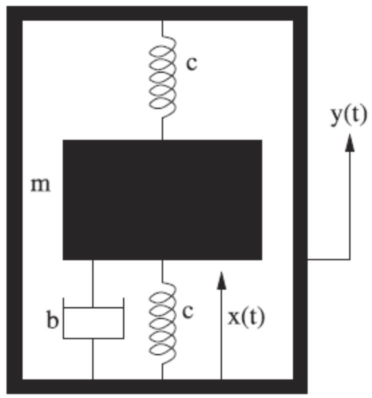
- **I**nertial **M**easurement **U**nit
  - Gyroscope: Angular velocity
  - Accelerometer: Linear Accelerations



Inertial Measurement Unit
3 accelerometers, 3 gyroscopes



Mechanical Gyroscope



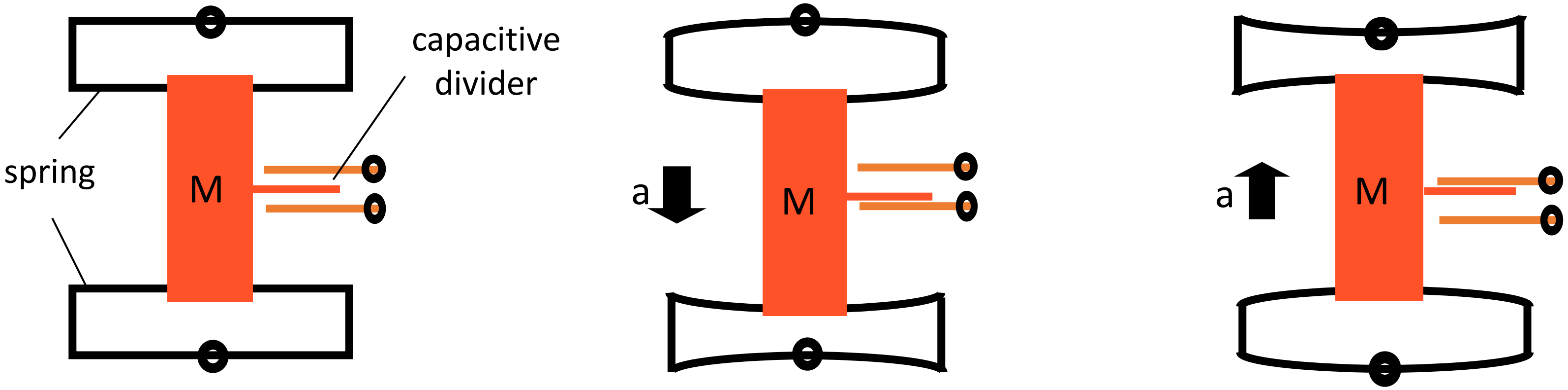
Mechanical Accelerometer

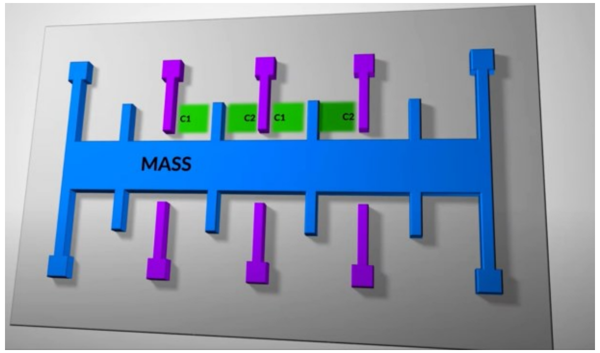
What is an IMU?

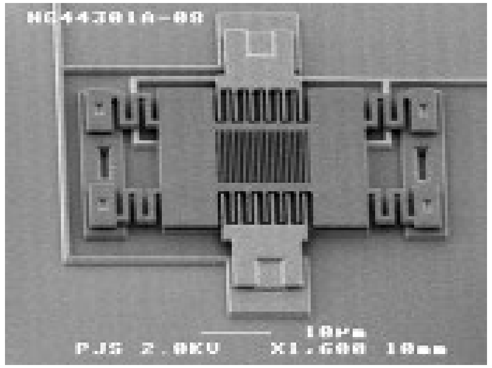
- Different categories
  - Mechanical ($100,000-1M)
  - Optical ($20,000-100k)
  - MEMS (from 1$ (phones) to 1,000$
    (higher cost because they have a microchip
    running a Kalman filter)
- For small mobile robots & drones:
  MEMS IMU are mostly used
  - Cheap
  - Power efficient
  - Light weight and solid state

MEMS Accelerometer

A spring-like structure connects the device to a seismic mass vibrating in a capacitive divider. A capacitive divider converts the displacement of the seismic mass into an electric signal. Damping is created by the gas sealed in the device.







MEMS Gyroscopes

- MEMS gyroscopes measure the **Coriolis forces** acting on MEMS vibrating structures (tuning forks, vibrating wheels, or **resonant solids**)
- Their working principle is similar to the haltere of a fly
- Haltere are small structures of some two-winged insects, such as flies. They are flapped rapidly and function as gyroscopes, informing the insect about rotation of the body during flight.

Why do we need an IMU?

- Monocular vision is **scale ambiguous** (Lecture 8, slide 7)
- Pure vision is **not robust enough**
  - Underexposure or overexposure (caused by low Dynamic Range)
  - Motion blur
  - Low texture
  - Not enough overlap between consecutive frames

Robustness is a critical issue: **Tesla accident, 2016**:

> *"**The autopilot** sensors on the Model S **failed** to **distinguish** a **white tractor-trailer** crossing the highway **against a bright sky**. "* [[The Guardian]]

Overexposure

Motion blur

Why is an IMU alone not enough?

- **Pure IMU integration will lead to large drift** (especially in cheap IMUs)
- Example: **1D scenario**. Double integration of acceleration returns the position:

$$x(t) = x_0 + v_0(t - t_0) + \iint_{t_0}^{t} a(\tau)d\tau^2$$

- If there is a **constant bias in the acceleration**, the error of position will be **proportional to t^2**
- Similarly for the orientation: if there is a **bias in angular velocity**, the error is **proportional to the time t**

GRADE/TIME	1 s	10 s	60 s	10 min	1 hr
Consumer	6 cm	6.5 m	400 m	200 km	39,000 km
Industrial	6 mm	0.7 m	40 m	20 km	3,900 km
Tactical	1 mm	8 cm	5 m	2 km	400 km
Navigation	<1 mm	1 mm	50 cm	100 m	10 km

Automotive,
smartphones,
and drones accelerometers

Table from Vectornav, one of the best IMU companies. Errors were computed assuming the device at rest:
https://www.vectornav.com/resources/inertial-navigation-primer/specifications--and--error-budgets/specs-inserrorbudget

Why visual inertial fusion?

- IMU and vision are complementary

Cameras	**IMU**
• **Exteroceptive sensor**: measures light energy from the environment	• **Proprioceptive sensor**: measures values internal to the system
× Sensitive to motion blur, HDR, texture	✓ Insensitive to motion blur, HDR, texture
✓ Drift is bounded when motion is bounded	× Drift grows unbounded regardless of the environment
✓ Precise in slow motion	× Less precise in slow motion (low signal-to-noise ratio)
× Limited output rate (~100 Hz)	✓ High output rate (1,000-10,000 Hz)
× Scale ambiguity in monocular setup	✓ No scale ambiguity: measurements are in absolute scale
	✓ Can be used as a prior to predict next feature positions

- What cameras and IMU have in common: both can be used to estimate the pose incrementally; this is known as dead-reckoning but suffers from drift over time. **Solution: fuse them together to reduce drift (see later)**
- IMUs can help **reduce the drift of VO** by up to a **factor of 10**.

Outline

- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

IMU Measurement Model

The model measures the **angular velocity** $\tilde{\omega}_B(t)$ and **acceleration** $\tilde{a}_B(t)$ **vectors** in the body frame B:

$$\boxed{\begin{aligned}\tilde{\omega}_B(t)\\ \tilde{a}_B(t)\end{aligned}} \begin{aligned} &= \omega_B(t) + \boxed{b^G(t) + n^G(t)} \\ &= R_{BW}(t)(a_W(t) - g) + \boxed{b^A(t) + n^A(t)}\end{aligned}$$

IMU biases + noise in body frame

Raw IMU measurements

(i.e., what you read from the sensor)

true $\boldsymbol{\omega}$ (in body frame) and true **a** (in world frame) to estimate

Notation:

- The superscript $()^G$ stands for gyroscope and $()^A$ for accelerometer
- R_{BW} is the rotation of the World frame W with respect to Body frame B
- The gravity vector g is expressed in the World frame
- Biases and noise are expressed in the body frame

What does an IMU measure during:

- free fall?
- in static conditions?

IMU Noise and Bias Model

- Additive, zero-mean Gaussian white noise: $n^G(t), n^A(t)$
- Biases: $b^G(t), b^A(t)$
  - The gyroscope and accelerometer biases are considered **slowly varying "constants"**. Their temporal fluctuation is modeled assuming that the derivative of the bias is a zero-mean Gaussian noise with standard deviation σ_b

$$\dot{\mathbf{b}}(t) = \sigma_b \mathbf{w}(t) \qquad \mathbf{w}(\mathrm{t}) \sim \mathbf{N}(0,1)$$

- Some facts about IMU biases:
- They change with **temperature** and **mechanical and atmospheric pressure**
- Thus, they **may also be different every time the IMU is turned on**
- Good news: **they can be estimated!** (see later)

Trawny, Roumeliotis, Indirect Kalman filter for 3D attitude estimation. Technical Report, University of Minnesota, 2005. PDF.
More info on the noise model: https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model

IMU Integration Model

- The **IMU Integration Model** computes the position, orientation, and velocity of the IMU in the world frame. To do this, we must first compute the acceleration $a(t)$ in the world frame from the measured one $\tilde{a}(t)$ in the body frame (see Slide 13):

$$a(t) = R_{WB}(t)\big(\tilde{a}(t) - b(t)\big) + g$$

- The position p_k at time t_k can then be **predicted** from the position p_{k-1} at time t_{k-1} by integrating all the inertial measurements $\{\tilde{a}_j, \tilde{\omega}_j\}$ within that time interval:

$$p_k = p_{k-1} + v_{k-1}(t_k - t_{k-1}) + \iint_{t_{k-1}}^{t_k} \big(R_{WB}(t)\big(\tilde{a}(t) - b(t)\big) + g\big)dt^2$$

NB:

- The rotation R_{WB} is computed from the gyroscope
- p_k depends on initial position and velocity. How do we measure them?

A similar expression can be obtained to predict the velocity v_k and orientation R_{WB} of the IMU in the world frame as functions of both $\tilde{a}_j$ and $\tilde{\omega}_j$

Trawny, Roumeliotis, Indirect Kalman filter for 3D attitude estimation. Technical Report, University of Minnesota, 2005. PDF.
More info on the noise model: https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model

IMU Integration Model

For convenience, the **IMU Integration Model** is normally written as

$$\begin{pmatrix} p_k \\ q_k \\ v_k \end{pmatrix} = f\begin{pmatrix} p_{k-1} \\ q_{k-1}, u \\ v_{k-1} \end{pmatrix} \quad \text{or, more compactly:} \quad \boxed{x_k = f(x_{k-1}, u)}$$

where:

- $x = \begin{bmatrix} p \\ q \\ v \end{bmatrix}$ represents the IMU **state**, i.e., **position, orientation, and velocity**
- q is the IMU **orientation** $\boldsymbol{R_{WB}}$ (usually represented using quaternions)
- $u = \{\tilde{a}_j, \tilde{\omega}_j\}$ are the accelerometer and gyroscope measurements in the time interval $[t_{k-1}, t_k]$

Trawny, Roumeliotis, Indirect Kalman filter for 3D attitude estimation. Technical Report, University of Minnesota, 2005. PDF.
More info on the noise model: https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model

Outline

- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

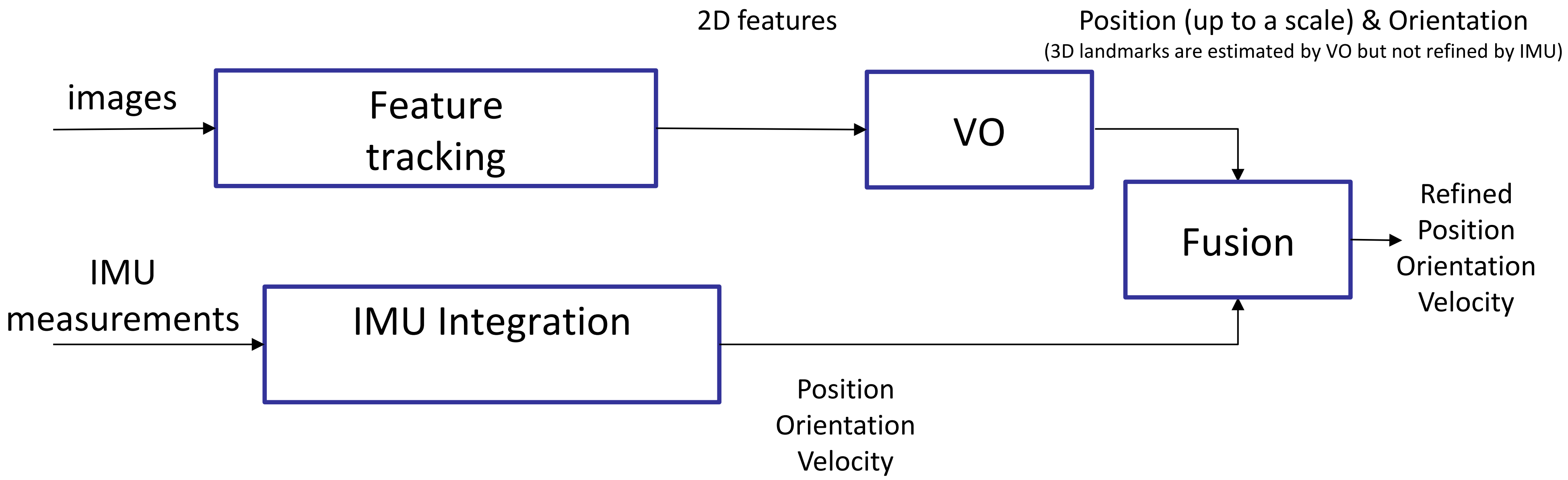
Visual Inertial Odometry

Different paradigms exist:

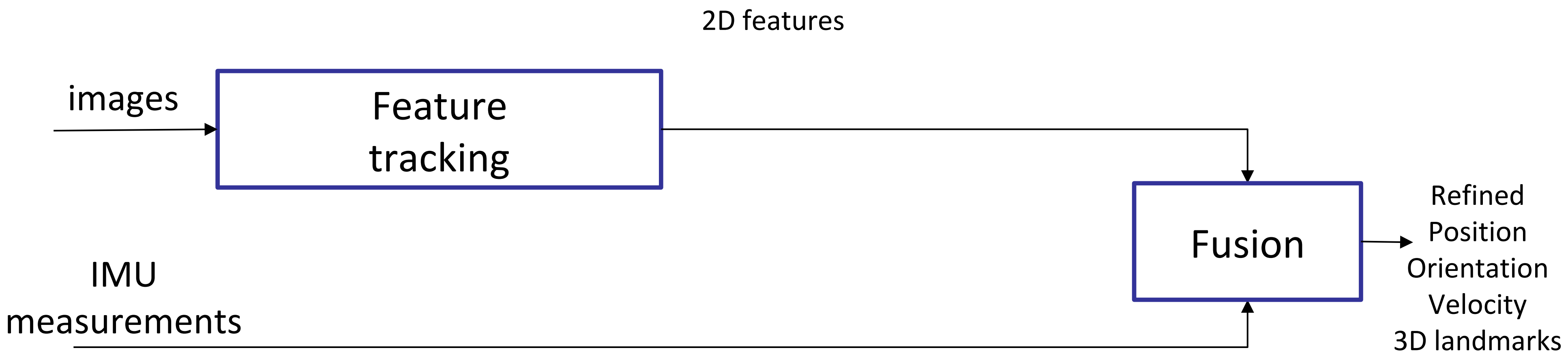
- **Loosely coupled**:
  - Treats **VO and IMU as two separate black boxes** (not coupled)
    - Each black box estimates pose and velocity from visual (up to a scale) and inertial data (absolute scale)
    - Easy to implement
    - Inaccurate. Should not be used if possible
- **Tightly coupled**:
  - Makes use of the **raw sensors' measurements** (2D features and IMU readings)
    - More accurate
    - More implementation effort

In this lecture, we will only see **tightly coupled approaches**

The **Loosely Coupled** Approach



The **Tightly Coupled** Approach



Filtering: Visual Inertial Formulation

- System states:

Tightly coupled: $\mathrm{X} = [p(t);\ q(t);\ v(t);\ \mathrm{b}^A(t);\ \mathrm{b}^G(t);\ \boxed{L_1;\ L_2; \ldots;\ L_k]}$

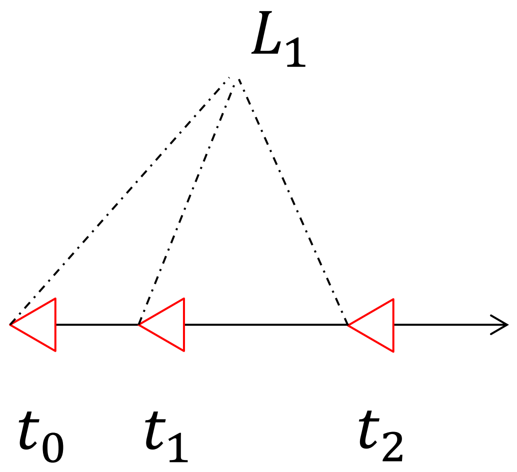
Loosely coupled: $\mathrm{X} = [p(t);\ q(t);\ v(t);\ \mathrm{b}^A(t);\ \mathrm{b}^G(t)]$

Corke, Lobo, Dias, *An Introduction to Inertial and Visual Sensing*, International Journal of Robotics Research (IJRR), 2007. PDF.

Outline

- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

Closed-form Solution (1D case)



- From a **single camera** we only get the relative position $\tilde{x}$ up to an **unknown scale factor s**, thus the absolute position x is:

$$x = s\tilde{x}$$

- From the IMU

$$x = x_0 + v_0(t_1 - t_0) + \iint_{t_0}^{t_1} a(t)dt^2$$

- By equating them

$$s\tilde{x} = x_0 + v_0(t_1 - t_0) + \iint_{t_0}^{t_1} a(t)dt^2$$

As shown in [Martinelli'14], if we assume to know x_0 (usually we set it to 0), then, even for 6DOF motion, **both s and v_0 can be determined in closed form** from a **single feature observation and 3 views**

Martinelli, *Closed-form solution of visual-inertial structure from motion*, International Journal of Computer Vision (IJCV), 2014. PDF.

Outline

- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

Non-linear Optimization Methods

VIO is solved as a non-linear Least Square optimization problem over:

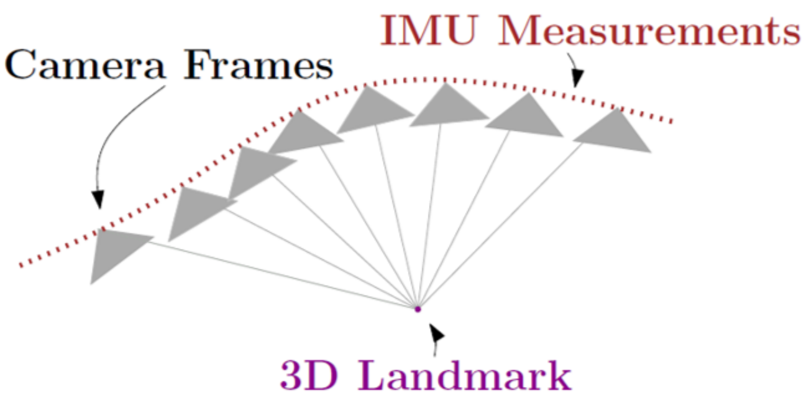
$$\{X, L, b^A, b^G\} = argmin_{\{X,\, L,\, b^A,\, b^G\}} \left\{ \sum_{k=1}^{N} \|f(x_{k-1}, u) - x_k\|^2_{\Lambda_k} + \sum_{k=1}^{N} \sum_{i=1}^{M} \left\|\pi(x_k, L^i) - z_k^i\right\|^2_{\Sigma_k^i} \right\}$$

IMU residuals

Reprojection residuals
(Bundle Adjustment term)

NB: it also optimizes the biases

Which initial guess do we use for the state and the biases?



[1] Jung, Taylor, *Camera Trajectory Estimation using Inertial Sensor Measurements and Structure from Motion Results*, International Conference on Computer Vision and Pattern Recognition (CVPR), 2001. PDF.
[2] Sterlow, Singh, *Motion estimation from image and inertial measurements*, International Journal of Robotics Research (IJRR), 2004. PDF.

Non-linear Optimization Methods

VIO is solved as a non-linear Least Square optimization problem over:

$$\{X, L, b^A, b^G\} = argmin_{\{X,\, L,\, b^A,\, b^G\}} \left\{ \sum_{k=1}^{N} \|f(x_{k-1}, u) - x_k\|_{\Lambda_k}^2 + \sum_{k=1}^{N} \sum_{i=1}^{M} \left\|\pi(x_k, L^i) - z_k^i\right\|_{\Sigma_k^i}^2 \right\}$$

IMU residuals

Reprojection residuals
(Bundle Adjustment term)

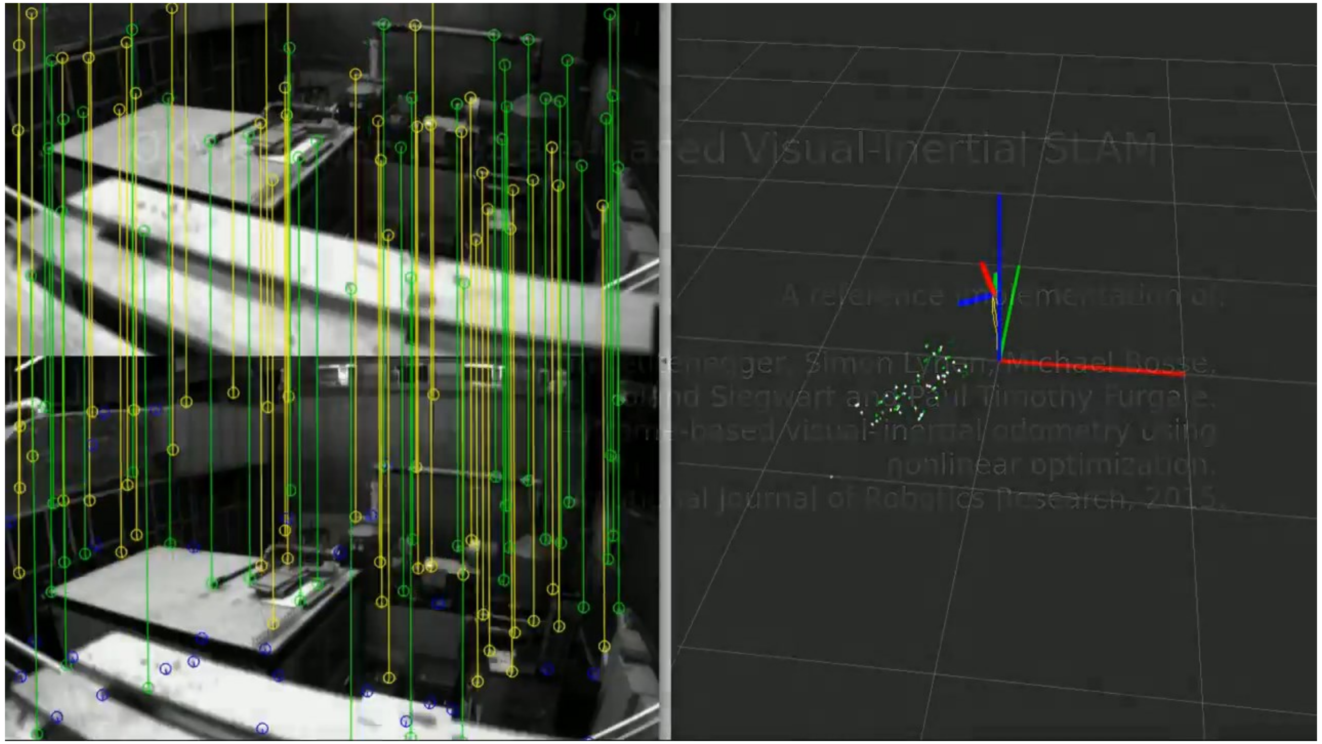
where

- $X = \{x_1, \ldots x_N\}$: set of **state estimates** x_k (**position, velocity, orientation**) at frame times k
- $L = \{L_1, \ldots, LM\}$: **3D landmarks**
- $f(x_{k-1}, u)$: **state prediction** obtained by integrating IMU measurements $u = \{\tilde{a}_j, \widetilde{\omega}_j\}$
- $\pi(x_k, l_i)$: **expected measurement at state estimates** x_k from projection of landmark L_i onto camera frame I_k
- z_{i_k}: **observed features**
- Λ_k: **inverse of the state covariance** from the IMU integration $f(x_{k-1}, u)$
- Σ_{i_k}: **inverse of the covariance** of the 2D feature position

[1] Jung, Taylor, *Camera Trajectory Estimation using Inertial Sensor Measurements and Structure from Motion Results*, International Conference on Computer Vision and Pattern Recognition (CVPR), 2001. PDF.
[2] Sterlow, Singh, *Motion estimation from image and inertial measurements*, International Journal of Robotics Research (IJRR), 2004. PDF.

Case Study 1: OKVIS

Because the complexity of the optimization is cubic with respect to the number of cameras poses and features (see Lecture 10, slide 32 and exercise 08), real-time operation becomes infeasible as the trajectory and the map grow over time, OKVIS proposed to only **optimize the current pose and a window of past keyframes**

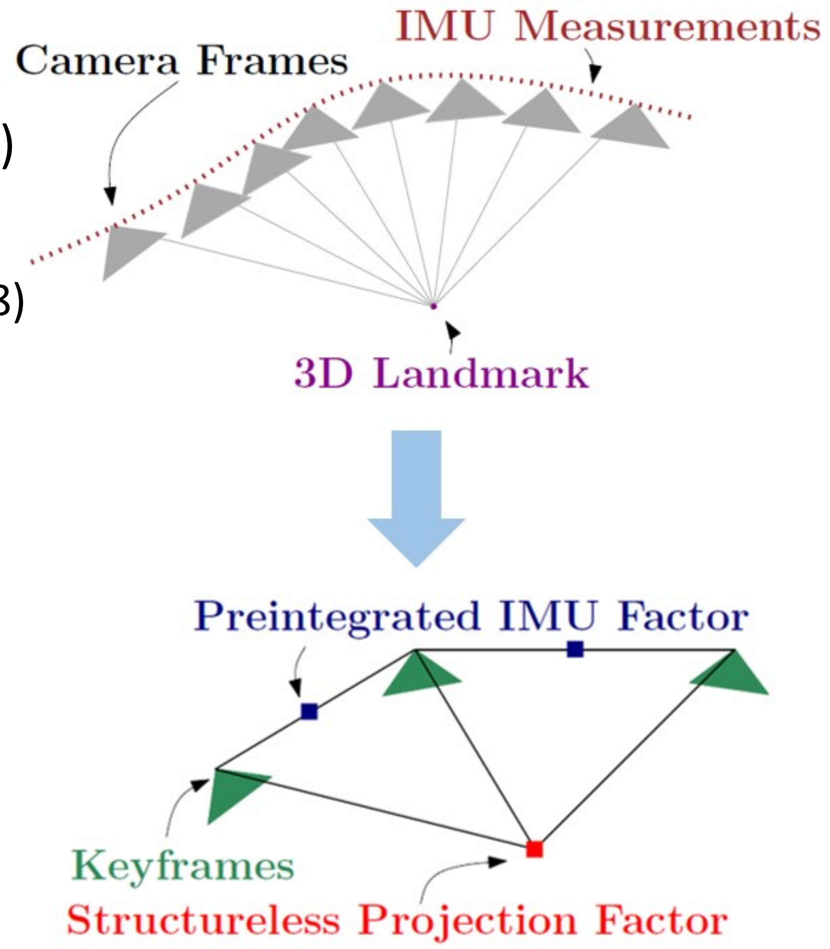


Leutenegger, Lynen, Bosse, Siegwart, Furgale, *Keyframe-based visual–inertial odometry using nonlinear optimization*, International Journal of Robotics Research (IJRR), 2015. [PDF](). [Video](). [Code]().

Case Study 2: SVO+GTSAM

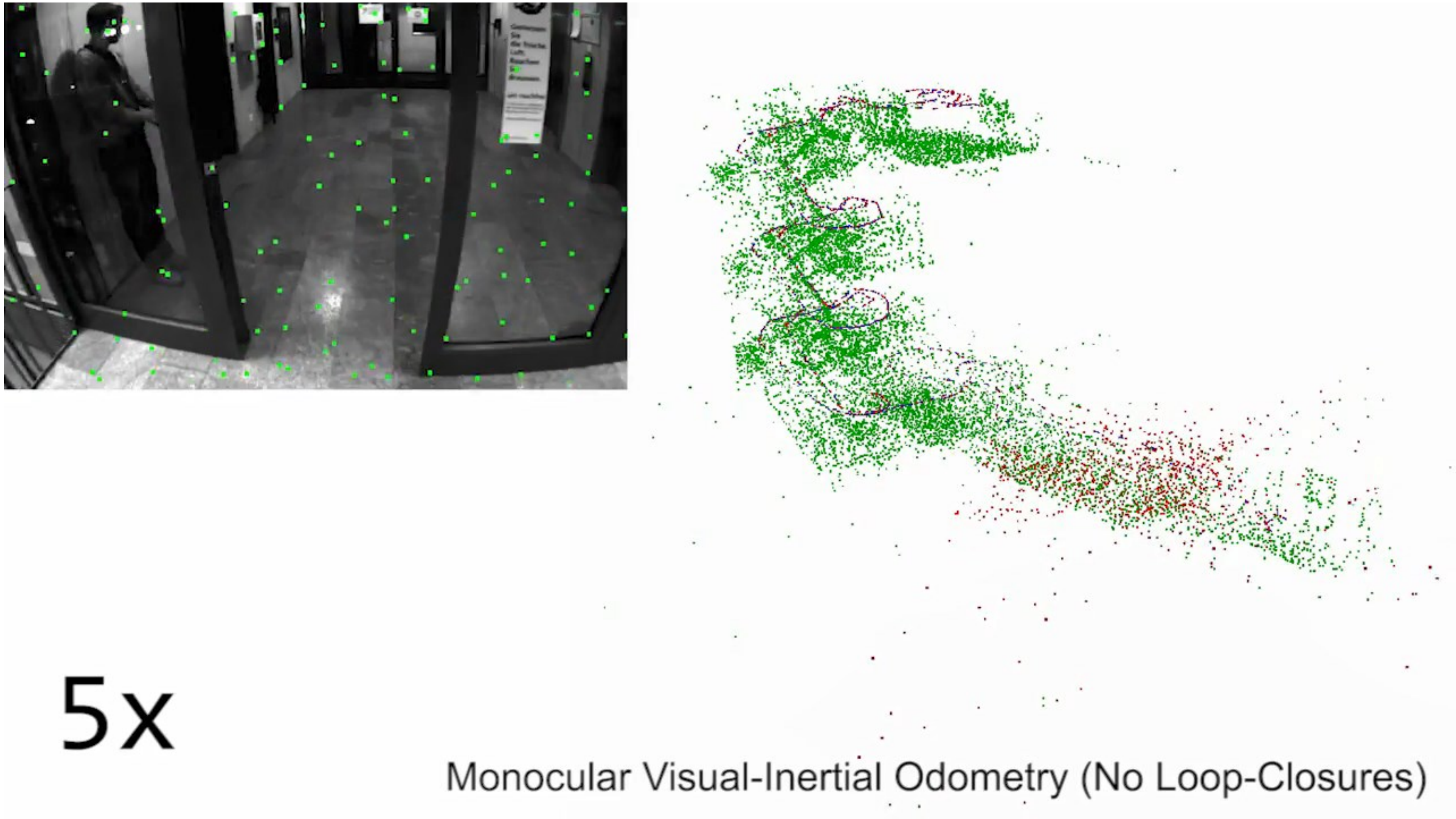
It solves the same optimization problem as OKVIS but:

- **It optimizes ALL keyframes** (from the start to the end of the trajectory)
- To make the optimization efficient
  - **Marginalizes 3D landmarks** (minimizes **Epipolar Line Distance** (Lecture 08) instead of the reprojection error)
  - **pre-integrates the IMU** data between keyframes (see later)
- **Optimization solved using Factor Graphs** via GTSAM
  - Very fast because it only **optimizes** the **poses** that are **affected by a new observation**



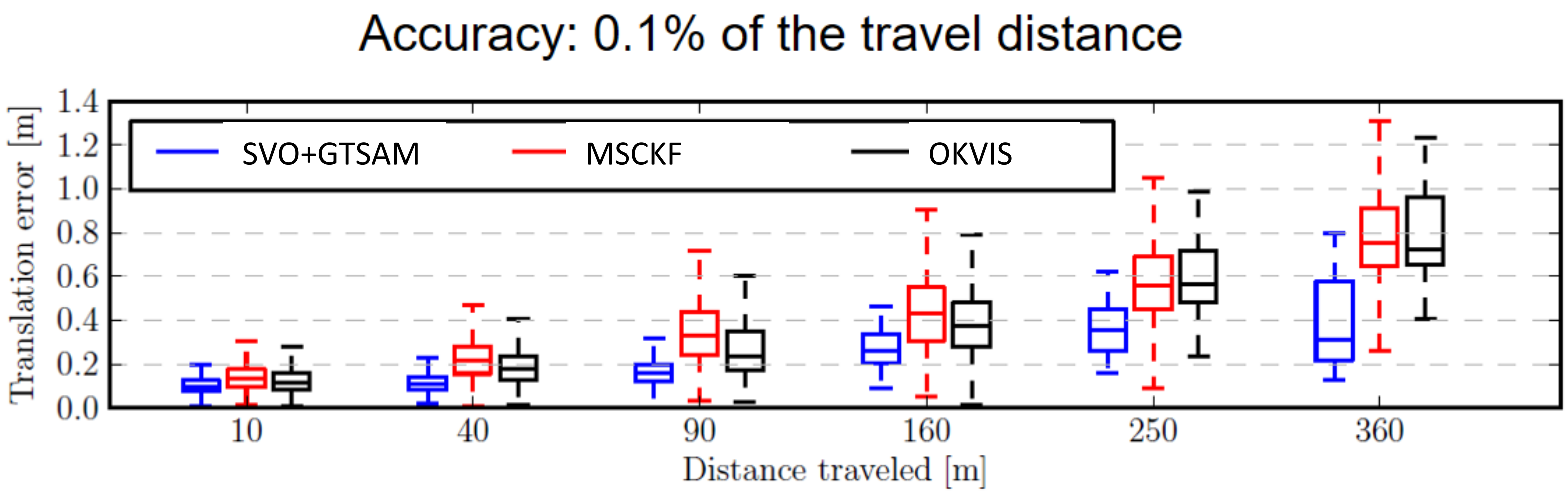
Forster, Carlone, Dellaert, Scaramuzza, *On-Manifold Preintegration for Real-Time Visual-Inertial Odometry*, IEEE Transactions on Robotics 2017. PDF. Video. Code. **Best Paper Award**.

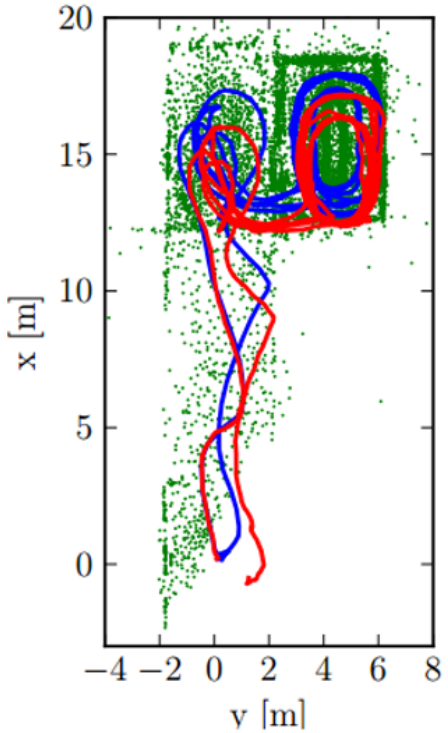
SVO+GTSAM



Forster, Carlone, Dellaert, Scaramuzza, *On-Manifold Preintegration for Real-Time Visual-Inertial Odometry*, IEEE Transactions on Robotics 2017. PDF. Video. Code. **Best Paper Award**.

SVO+GTSAM





Forster, Carlone, Dellaert, Scaramuzza, *On-Manifold Preintegration for Real-Time Visual-Inertial Odometry*, IEEE Transactions on Robotics 2017. PDF. Video. Code. **Best Paper Award**.

Problem with IMU integration

- The integration of IMU measurements, $f(x_{k-1}, u)$, from $k-1$ to k is related to the state estimation at time $k-1$
- During optimization, every time the linearization point at $k-1$ changes, the integration between $k-1$ and k must be re-evaluated, thus slowing down the optimization

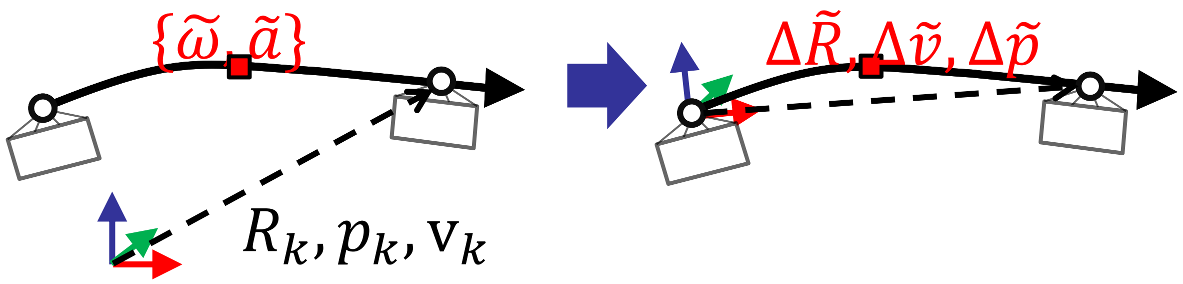
$$\{X, L, b^A, b^G\} = argmin_{\{X,\, L,\, b^A,\, b^G\}} \left\{ \underbrace{\sum_{k=1}^{N} \|f(x_{k-1}, u) - x_k\|^2_{\Lambda_k}}_{\textit{IMU residuals}} + \underbrace{\sum_{k=1}^{N} \sum_{i=1}^{M} \left\|\pi(x_k, L^i) - z_k^i\right\|^2_{\Sigma_k^i}}_{\substack{\textit{Reprojection residuals} \\ \text{(Bundle Adjustment term)}}} \right\}$$

- **Idea: Preintegration**
  - defines relative motion increments, expressed in body frame, which are independent on the global position, orientation, and velocity at k [1]
  - [2] uses this theory by leveraging the manifold structure of the rotation group SO(3)

[1] Lupton, Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions, IEEE Transactions on Robotics (T-RO), 2012. PDF.
[2] Forster, Carlone, Dellaert, Scaramuzza, *On-Manifold Preintegration for Real-Time Visual-Inertial Odometry*, IEEE Transactions on Robotics 2017. PDF. Video. Code.

IMU Pre-Integration

$$f(x_{k-1}, u) - x_k$$



Standard:
Evaluate **error in global frame**:

Repeats integration when previous state changes!

Preintegration:
Evaluate **relative errors (i.e., in body frame)**:

Preintegration of IMU deltas possible with **no initial condition required**.

Outline

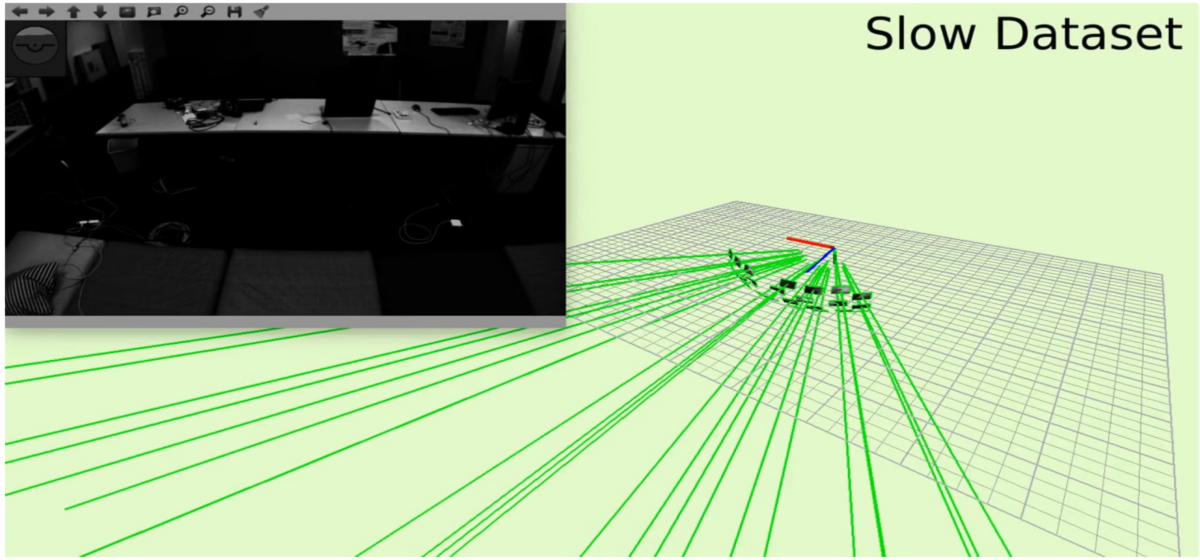
- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

Non-linear optimization vs. Filtering

Non-linear Optimization methods	Filtering methods
Optimize a **window** of multiple states (or all the states) using non-linear Least-Squares optimization	Solve the same problem by **running only one iteration of the optimization** function (e.g., using Extended Kalman Filter (EKF))
✓Multiple iterations (it re-linearizes at each iteration) ✓Achieves the highest accuracy ✓Slower	× One iteration only × Sensitive to linearization point ✓Fastest

Filter-based VIOs - Case Study 1: ROVIO

- EKF state: $X = [p(t);\ q(t);\ v(t);\ \mathrm{b}^a(t);\ \mathrm{b}^g(t);\ L_1;\ L_2; \ldots;\ L_k]$
- Basic idea:
  1. **Prediction step**: predicts next position, velocity, orientation, and features using IMU integration model
  2. **Measurement update**: refines state by leveraging visual constraint (ROVIO minimizes the photometric error between corresponding points (alternative would be the reprojection error))



Bloesch, Burri, Omari, Hutter, Siegwart, *Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback*, International Journal of Robotics Research (IJRR), 2017. PDF. Code.

ROVIO: Problems

- **Complexity** of the EKF grows **quadratically** in the number of estimated landmarks
  - Thus, **max 20 landmarks** are tracked to allow real-time operation

- **Only updates the most recent state**

Filter-based VIOs - Case Study 2: MSCKF

- **MSCKF** (**Multi-State Constraint Kalman Filter**) updates **multiple past poses** $\{p_{C_1}, q_{C_1}, \dots, p_{C_N}, q_{C_N}\}$ in addition to the current state $\{p(t), q(t), v(t)\}$. State vector:

$$X = \left[p(t);\ \ q(t);\ \ v(t);\ \mathrm{b}^A(t);\ \mathrm{b}^G(t);\ \ p_{C_1};\ \ q_{C_1};\ \ \dots;\ \ p_{C_N};\ \ q_{C_N}\right]$$

- **Prediction step:** same as ROVIO

- **Measurement update:**
  - Differently from ROVIO,
    - **landmark positions** are **not added to the state vector**, thus can run very fast independently of the number of features
    - Visual constraint is obtained from the **Epipolar Line Distance** (Lecture 08)

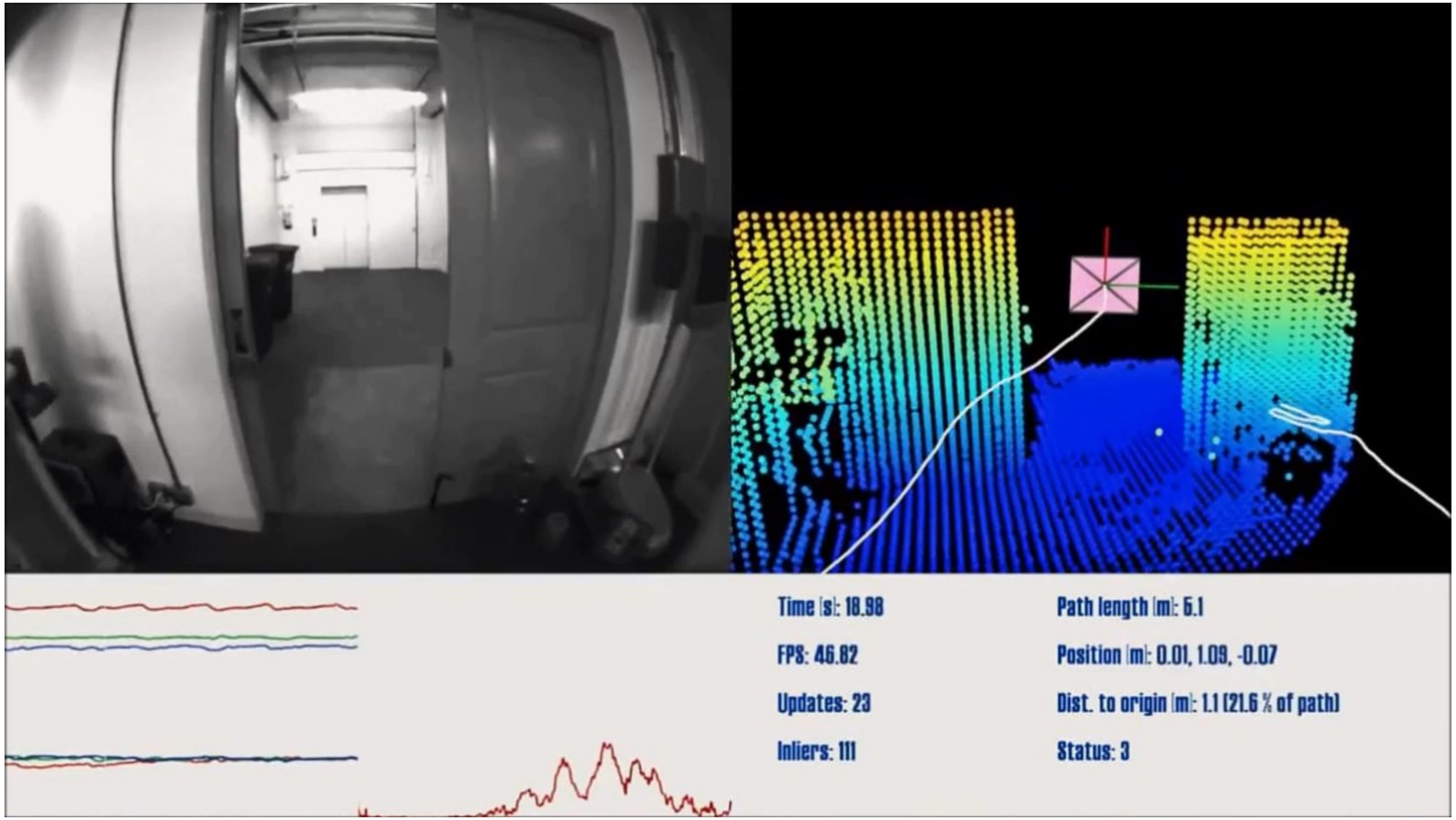
- Used in spacecraft landing (**NASA/JPL Moon and Mars landing**), **DJI drones, Google ARCore, Apple ARKit**

- Released open source within the OpenVins project: https://github.com/rpng/open_vins

[1] Mourikis, Roumeliotis, *A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation*, International Conference on Robotics and Automation (ICRA), 2007. PDF.

[2] Li, Mourikis, *High-precision, consistent EKF-based visual–inertial odometry*, International Journal of Robotics Research (IJRR), 2013. PDF.

MSCKF running in Google ARCore (former Google Tango)



Video

[1] Mourikis, Roumeliotis, *A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation*, International Conference on Robotics and Automation (ICRA), 2007. PDF.

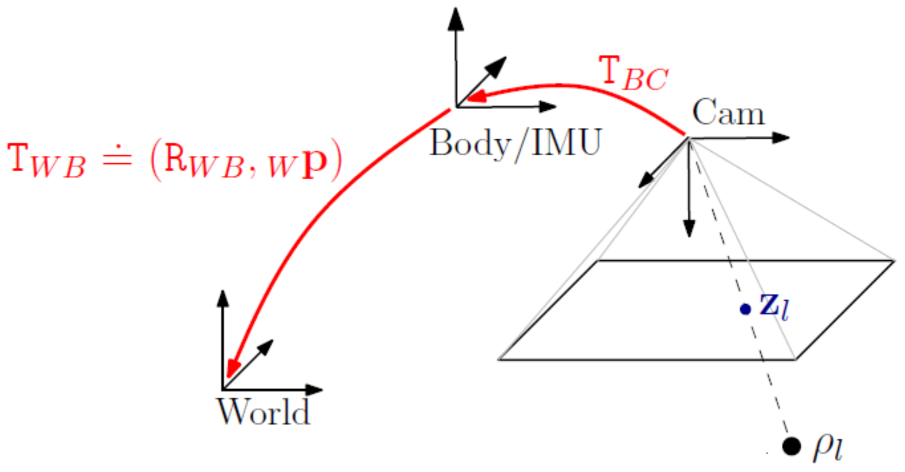
[2] Li, Mourikis, *High-precision, consistent EKF-based visual–inertial odometry*, International Journal of Robotics Research (IJRR), 2013. PDF.

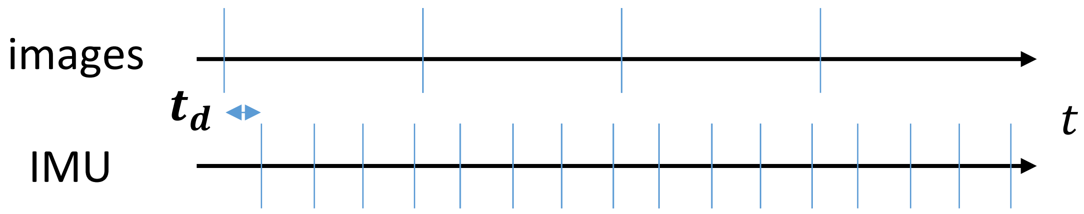
Outline

- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

Camera-IMU Calibration

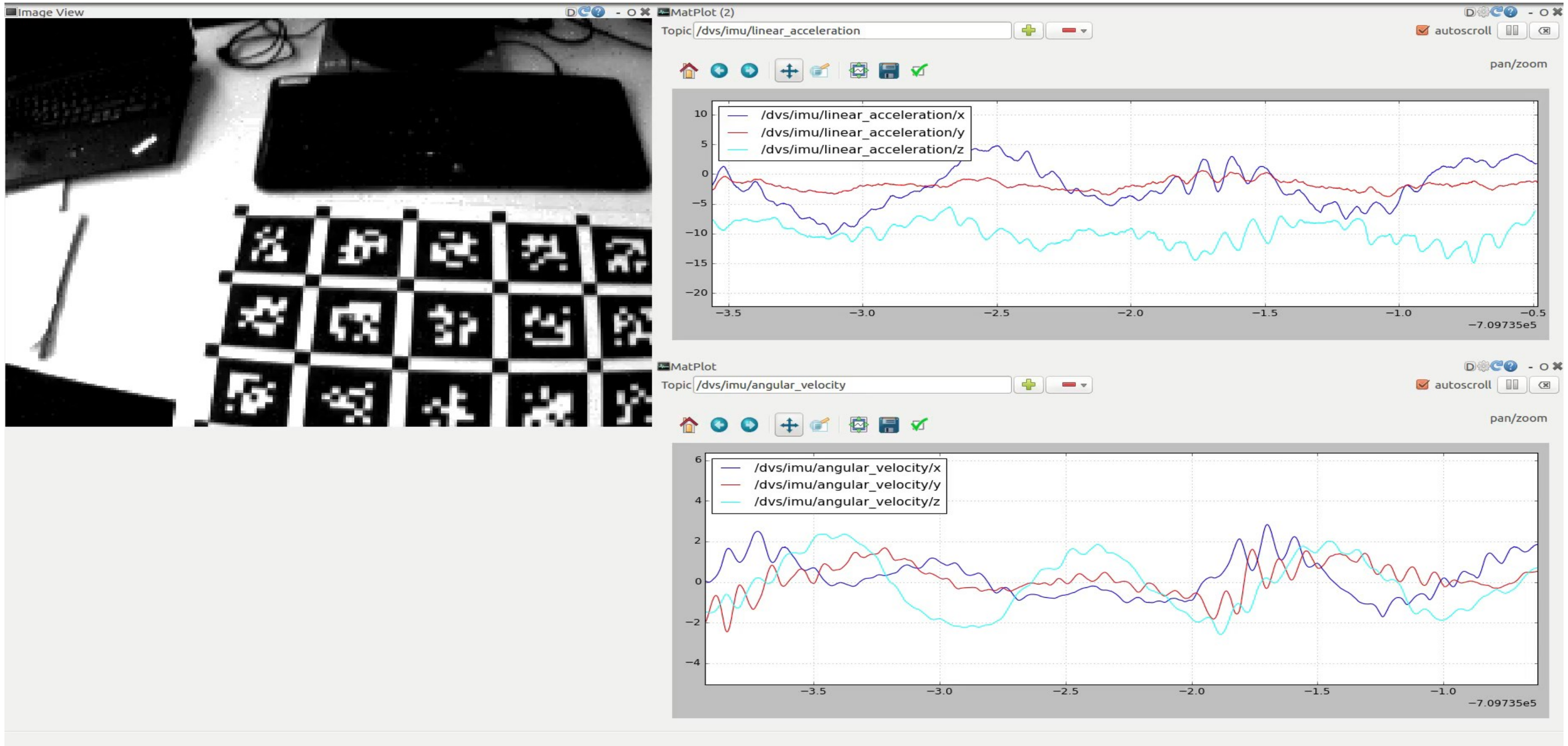
- **Goal**: estimate the **rigid-body transformation** $\boldsymbol{T_{BC}}$ and time **offset** $\boldsymbol{t_d}$ **between** the **camera** and the **IMU** caused by communication delays and the internal sensor delays (introduced by filters and logic).
- **Assumptions**: Camera and IMU rigidly attached. Camera intrinsically calibrated.
- **Data**:
  - Image points from calibration pattern (checkerboard or QR board)
  - IMU measurements: accelerometer $\{a_k\}$ and gyroscope $\{\omega_k\}$





Kalibr Toolbox

- Code: https://github.com/ethz-asl/kalibr/wiki/camera-imu-calibration



Furgale, Rehder, Siegwart, *Unified Temporal and Spatial Calibration for Multi-Sensor Systems*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013. PDF. Code.

Kalibr Toolbox

- Solves a non-linear Least Square optimization problem similar to that seen before but also optimizes over T_{BC}, t_d:

$$\{X, L, T_{BC}, t_d, b^a, b^g\} = argmin_{\{X, L, T_{BC}, t_d,\, b^a,\, b^g\}} \left\{ \sum_{k=1}^{N} \| f(x_{k-1}, u) - x_k \|^2_{\Lambda_k} + \sum_{k=1}^{N} \sum_{i=1}^{M} \| \pi(x_k, L^i) - z_k^i \|^2_{\Sigma_k^i} \right\}$$

IMU residuals

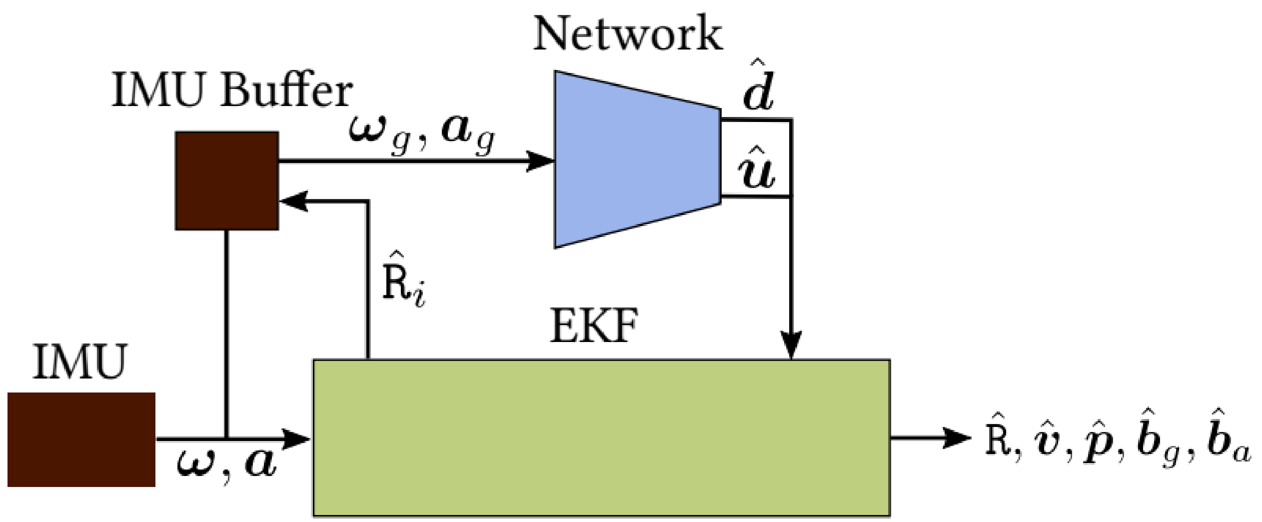
Reprojection residuals
(Bundle Adjustment term)

- Continuous-time modelling using splines for X
- Numerical solver: Levenberg-Marquardt

Furgale, Rehder, Siegwart, *Unified Temporal and Spatial Calibration for Multi-Sensor Systems*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013. PDF. Code.

Latest and Greatest ☺

TLIO: Learned Inertial Odometry for Pedestrians

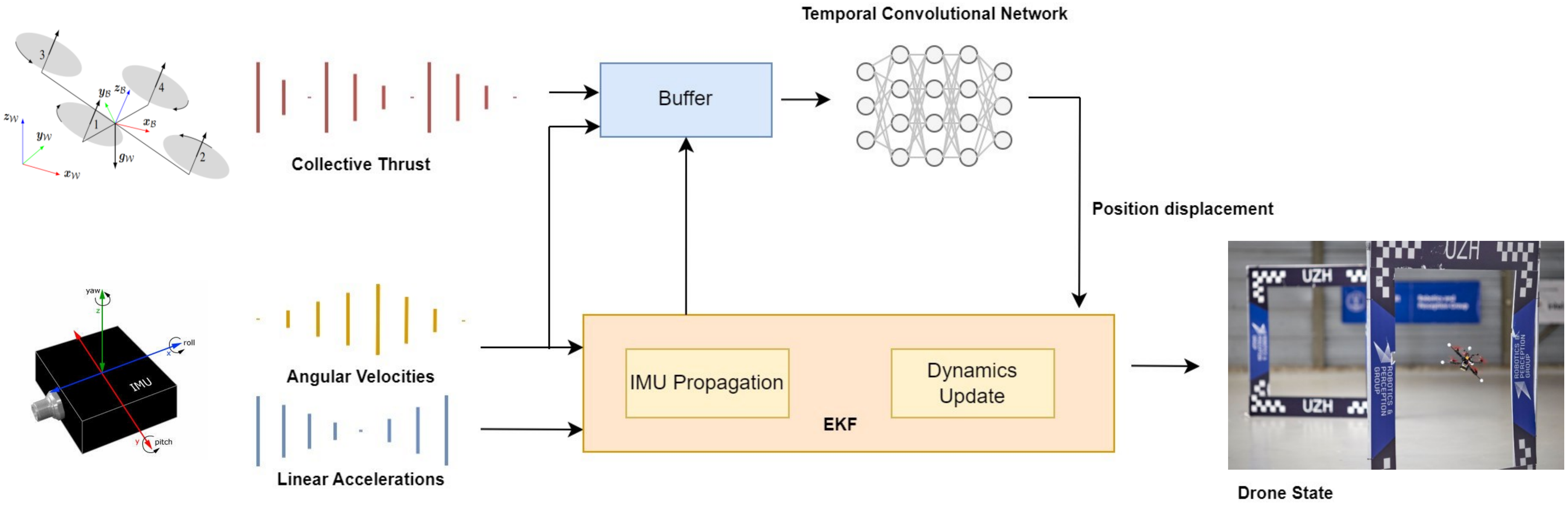


- IMU-only odometry for pedestrians combining deep learning with an extended Kalman filter (EKF)
- A neural network regresses position displacement and its uncertainty from a window of the most recent IMU measurements
- The position displacement is then fused into an EKF to estimate the pose, velocity, and bias of the IMU.
- Enables robust state estimation in challenging environments for visual frontends, e.g. high dynamic scenes, low light, etc.

[1] Liu, Caruso, Ilg, Dong, Mourikis, Daniilidis, Kumar, Engel, TLIO: Tight Learned Inertial Odometry, Robotics and Automation Letters (RA-L), 2020.

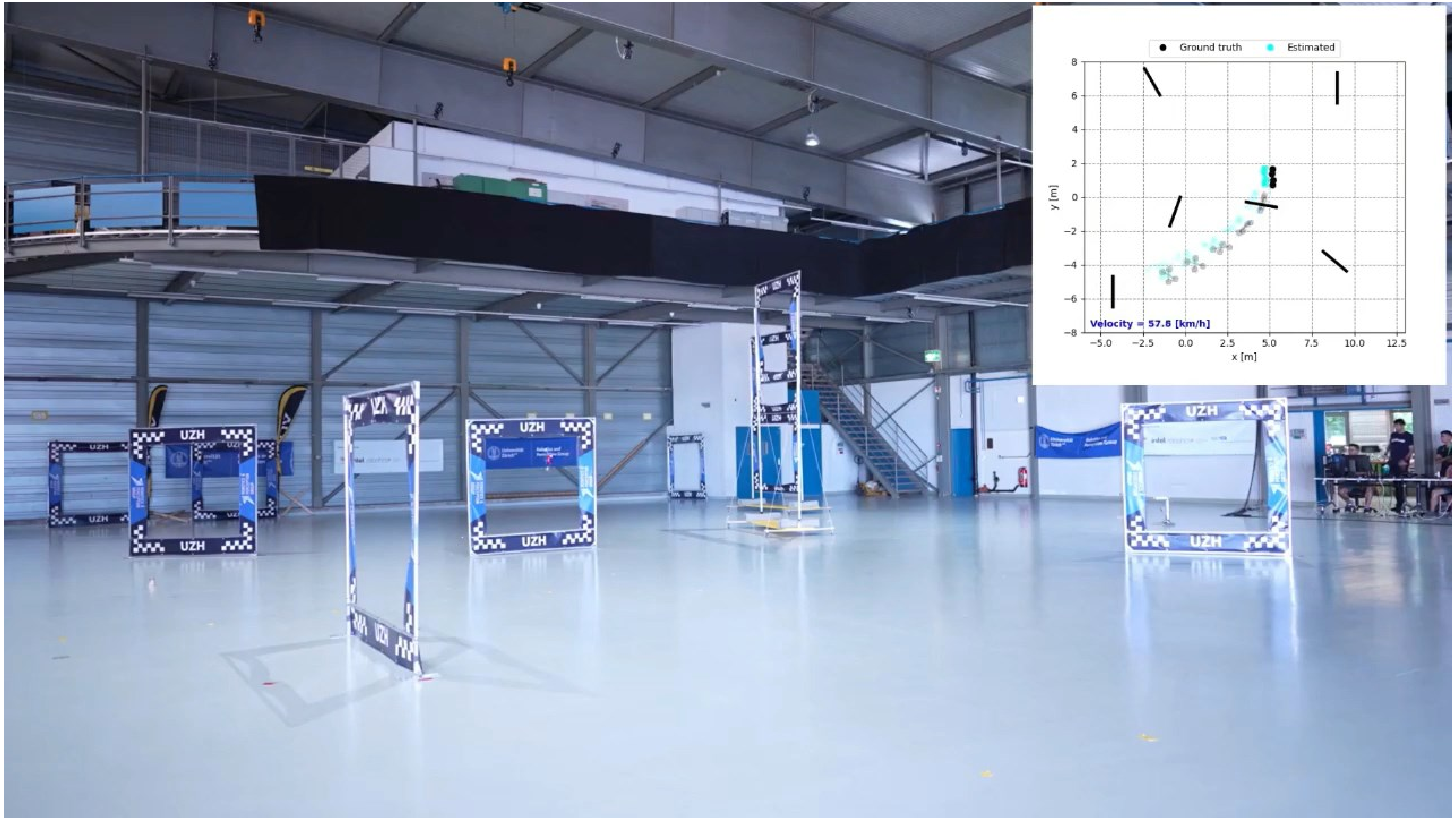
Learned Inertial Odometry

- We propose a learning-based odometry algorithm that **uses an IMU as the only sensor modality** for autonomous drone racing
- The core idea is to couple a model-based filter, driven by the IMU measurements, with a **learning-based drone dynamics model**



Cioffi, Bauersfeld, Kaufmann, Scaramuzza, *Learned Inertial Odometry for Autonomous Drone Racing*, RA-L, 2023

Learned Inertial Odometry



Cioffi, Bauersfeld, Kaufmann, Scaramuzza, *Learned Inertial Odometry for Autonomous Drone Racing*, RA-L, 2023

Readings

- Scaramuzza, Zhang, **Visual-Inertial Odometry of Aerial Robots**, Encyclopedia of Robotics, Springer, 2019, PDF.

- Huang, **Visual-inertial navigation: A concise review**, International Conference on Robotics and Automation (ICRA), 2019. PDF.

- Corke, Lobo, Dias, **An Introduction to Inertial and Visual Sensing**, International Journal of Robotics Research (IJRR), 2007. PDF.

Understanding Check

Are you able to answer the following questions?

- Why is it recommended to use an IMU for Visual Odometry?
- Why not just using an IMU and do inertial odometry (i.e., without a camera)?
- What is the basic idea behind MEMS IMUs?
- What is the drift of a consumer IMU?
- What is the IMU measurement model? (formula)
- What causes the bias in an IMU?
- How do we model the bias?
- How do we integrate the acceleration to get the position (formula)?
- What is the definition of loosely coupled and tightly coupled visual inertial fusion?
- How does non-linear optimization-based visual inertial odometry? Can you write down the cost function and illustrate its meaning?
- What does IMU-camera calibration do? Can you illustrate the unknowns and how to estimate them?