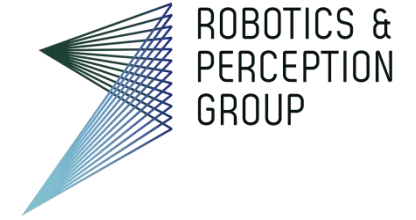




**University of  
Zurich** <sup>UZH</sup>



# Vision Algorithms for Mobile Robotics

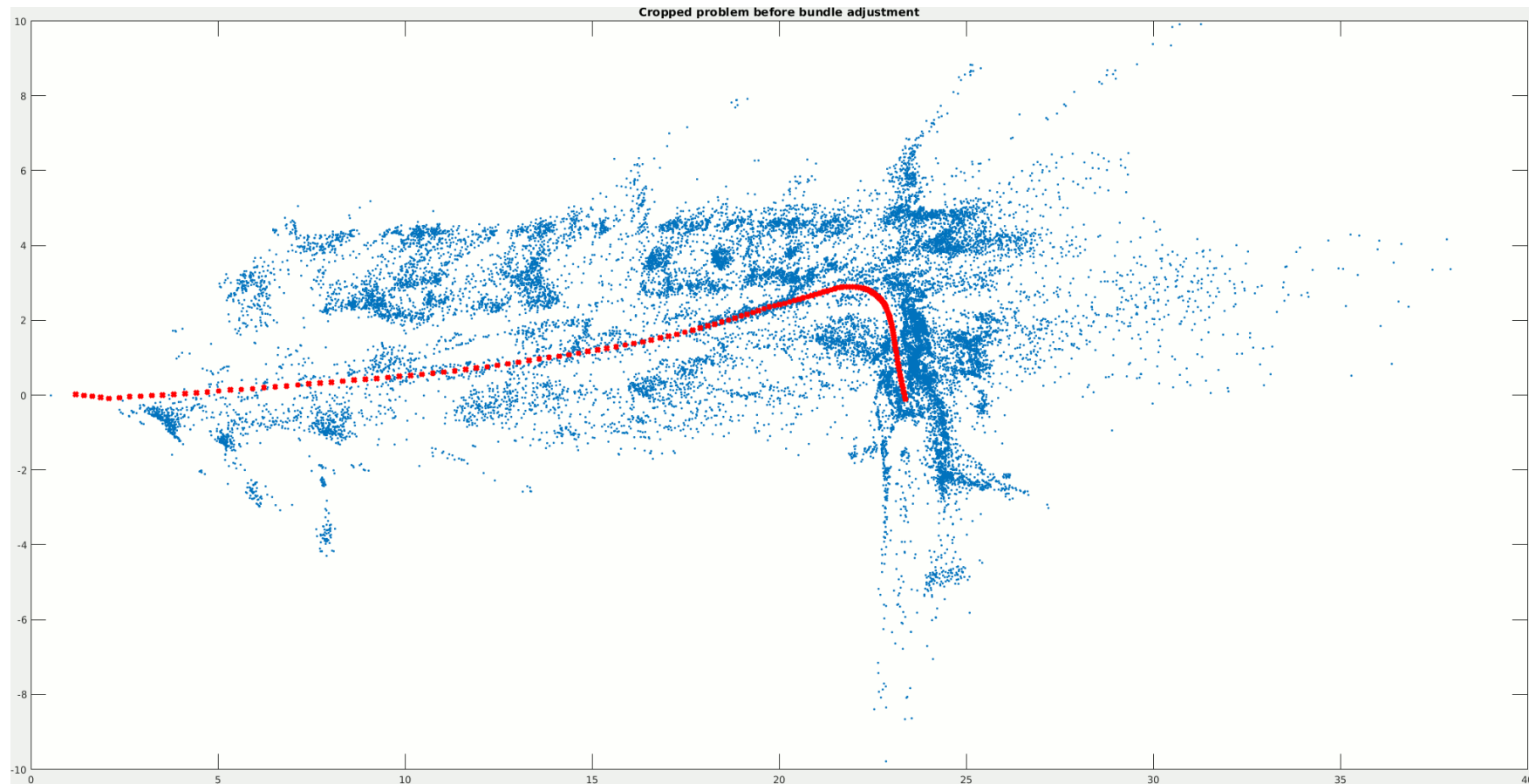
## Lecture 13 Visual Inertial Fusion

Davide Scaramuzza

<http://rpg.ifi.uzh.ch>

# Lab Exercise – Today

## Bundle Adjustment

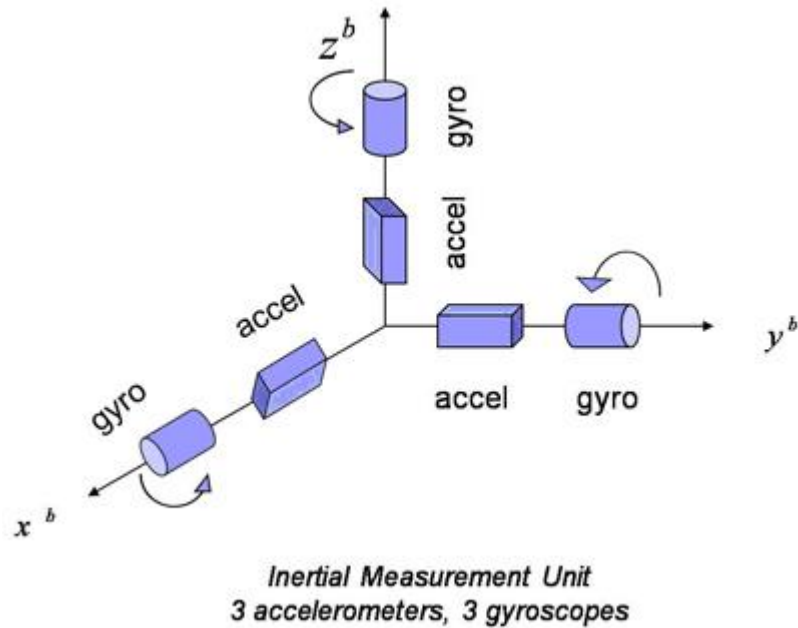


# Outline

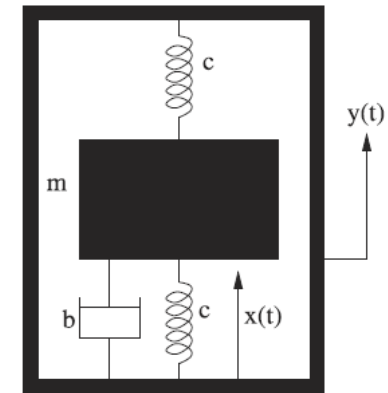
- What is an IMU and why do we need it?
- IMU model
- Visual Inertial Odometry (VIO)
  - Closed-form solution
  - Non-linear optimization methods
  - Filtering methods
- Camera-IMU extrinsic calibration and Synchronization

# What is an IMU?

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



Mechanical Gyroscope



Mechanical Accelerometer

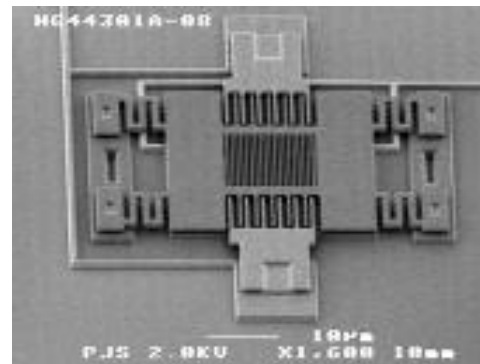
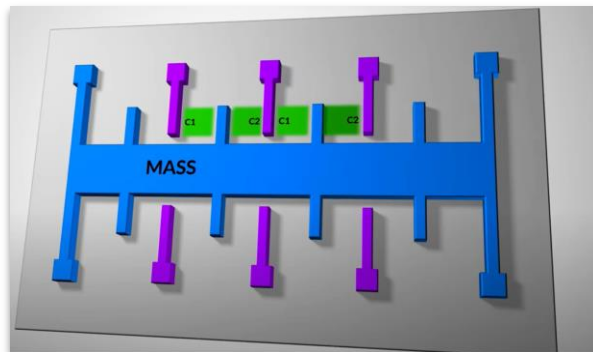
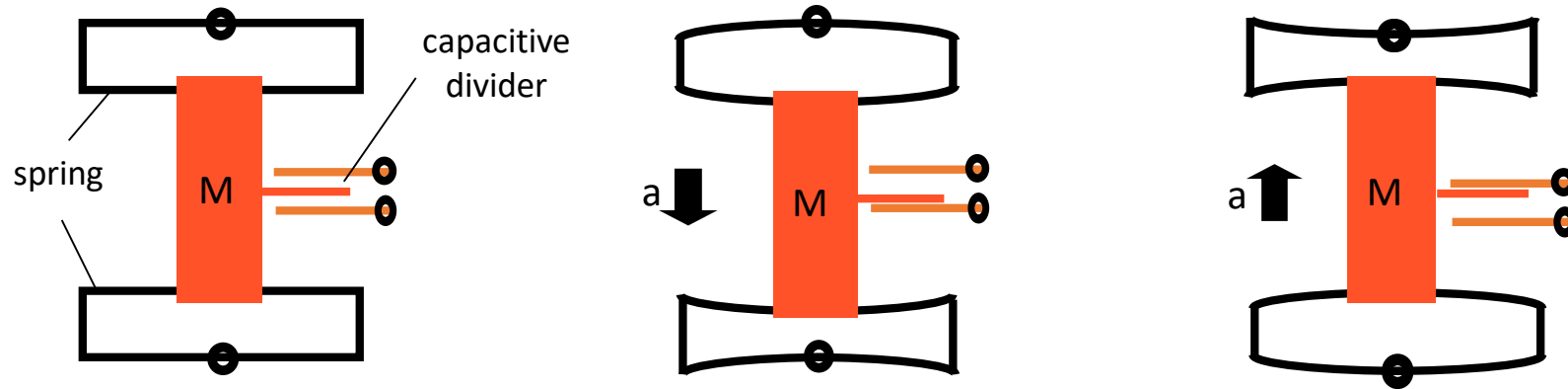
# What is an IMU?

- 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**
- Pure vision is **not robust enough**
  - Low texture
  - High Dynamic Range (HDR)
  - High speed motion

## Robustness is a critical issue: Tesla accident

*“The autopilot sensors on the Model S failed to distinguish a white tractor-trailer crossing the highway against a bright sky.” [[The Guardian](#)]*

Motion blur



High Dynamic Range





# 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

- **Exteroceptive sensor:** measures light energy from the environment
- × Sensitive to motion blur, HDR, texture
- ✓ Drift is bounded when observing the same scene
- ✓ Precise in slow motion
- × Limited output rate (~100 Hz)
- × Scale ambiguity in monocular setup

## IMU

- **Proprioceptive sensor:** measures values internal to the system
- ✓ Insensitive to motion blur, HDR, texture
- × Drift grows unbounded regardless of the environment
- × Less precise in slow motion (low signal-to-noise ratio)
- ✓ High output rate (1,000-10,000 Hz)
- ✓ 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 (known as dead-reckoning), which suffers from drift over time. **Solution: fuse them together to reduce drift (see later)**

# 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 angular velocity  $\tilde{\omega}_B(t)$  and acceleration  $\tilde{a}_B(t)$  in the body frame  $B$ :

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

Raw IMU measurements

IMU biases + noise in body frame

true  $\omega$  (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  $g$  is expressed in the World frame
- Biases and noise are expressed in the body frame

# 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 by saying that the derivative of the bias is a zero-mean Gaussian noise with standard deviation  $\sigma_b$

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

- 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)

# 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):

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

- 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:

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

- 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  $\tilde{\omega}_j$

# IMU Integration Model

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

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

where:

- $x = \begin{bmatrix} p \\ q \\ v \end{bmatrix}$  represents the IMU **state**, i.e., **position, orientation, and velocity**
- $q$  is the IMU **orientation**  $R_{WB}$  (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



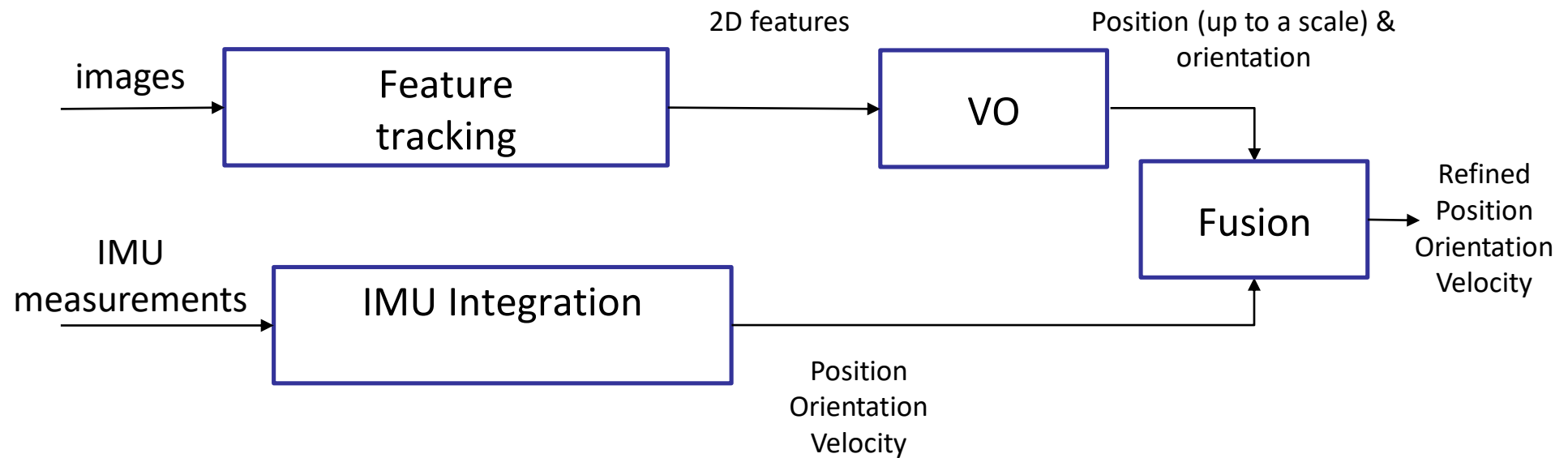
# Visual Inertial Odometry

Different paradigms exist:

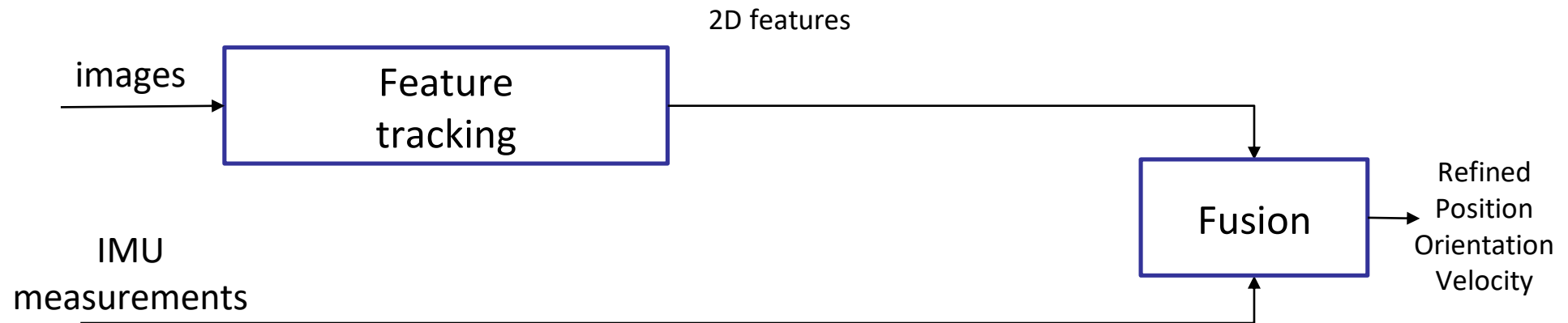
- **Loosely coupled:**
  - Treats **VO and IMU as two separate black boxes** (not coupled)
    - Each black box estimates pose and velocity from visual (up to a scale) and inertial data (absolute scale)
    - Easy to implement
    - Inaccurate. Should not be used
- **Tightly coupled:**
  - Makes use 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:**  $X = [p(t); q(t); v(t); b^a(t); b^g(t); L_1; L_2; \dots; L_k]$

**Loosely coupled:**  $X = [p(t); q(t); v(t); b^a(t); b^g(t)]$

# 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 absolute position  $x$  up to a **unknown scale factor  $s$** , thus

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

where  $\tilde{x}$  is the relative motion estimated by the camera.

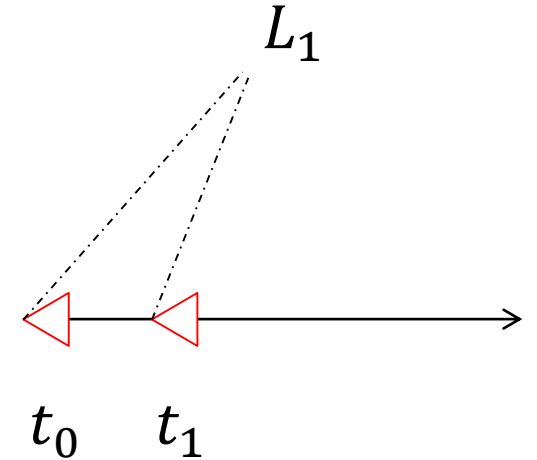
- From the IMU

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

- By equating them

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

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



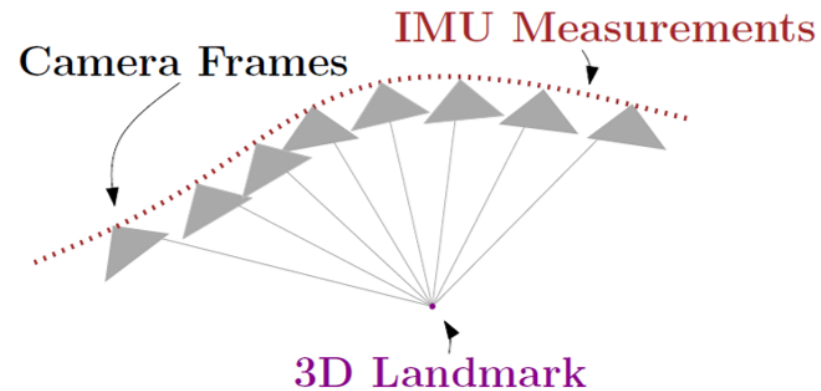
# Outline

- 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:

$$\{X, L, b^a, b^g\} = \underset{\{X, L, b^a, b^g\}}{\operatorname{argmin}} \left\{ \underbrace{\sum_{k=1}^N \|f(x_{k-1}, u) - x_k\|_{\Lambda_k}^2}_{\text{IMU residuals}} + \underbrace{\sum_{k=1}^N \sum_{i=1}^M \|\pi(x_k, L_i) - z_{i_k}\|_{\Sigma_{i_k}}^2}_{\substack{\text{Reprojection residuals} \\ \text{(Bundle Adjustment term)}}} \right\}$$



[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:

$$\{X, L, b^a, b^g\} = \underset{\{X, L, b^a, b^g\}}{\operatorname{argmin}} \left\{ \underbrace{\sum_{k=1}^N \|f(x_{k-1}, u) - x_k\|_{\Lambda_k}^2}_{\text{IMU residuals}} + \underbrace{\sum_{k=1}^N \sum_{i=1}^M \|\pi(x_k, L_i) - z_{i_k}\|_{\Sigma_{i_k}}^2}_{\substack{\text{Reprojection residuals} \\ \text{(Bundle Adjustment term)}}} \right\}$$

where

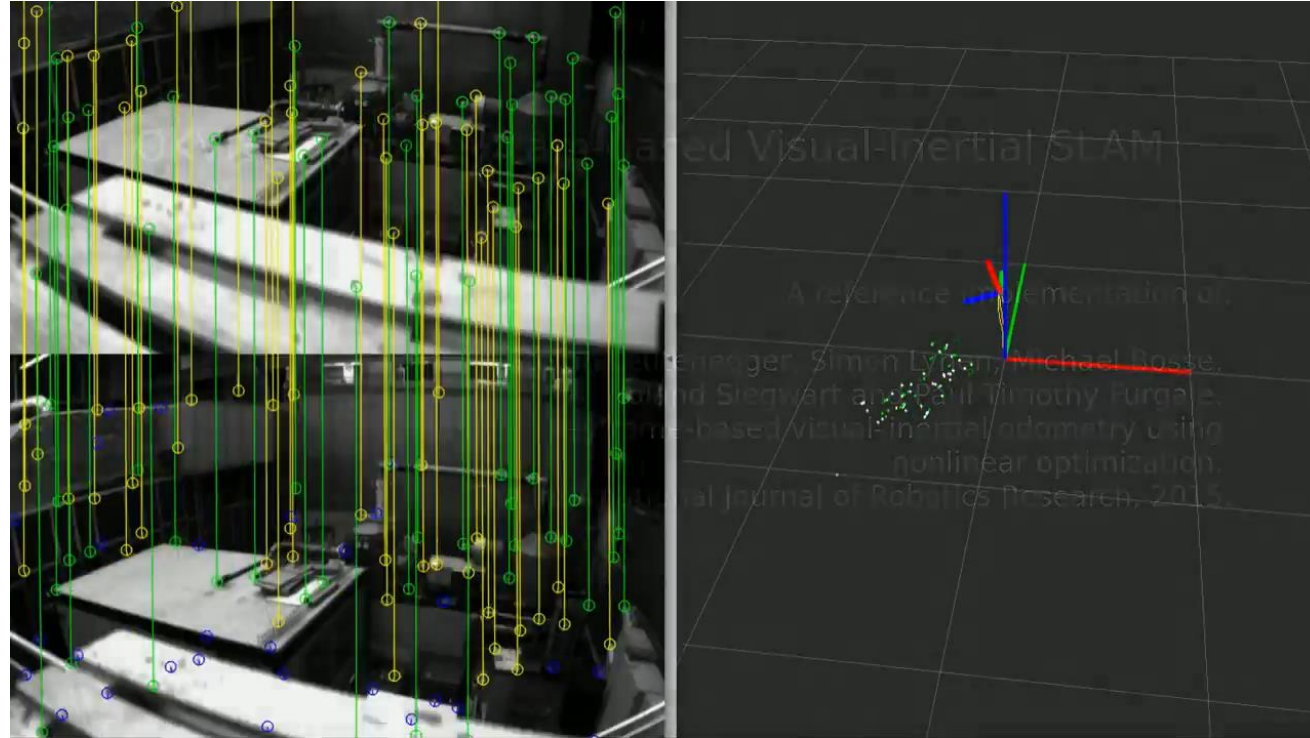
- $X = \{x_1, \dots, x_N\}$ : set of **state estimates**  $x_k$  (**position, velocity, orientation**) at frame times  $k$
- $L = \{L_1, \dots, L_M\}$ : **3D landmarks**
- $f(x_{k-1}, u)$ : **state prediction** obtained by integrating IMU measurements  $u = \{\tilde{a}_j, \tilde{\omega}_j\}$
- $\pi(x_k,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

[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 camera poses and features, 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**

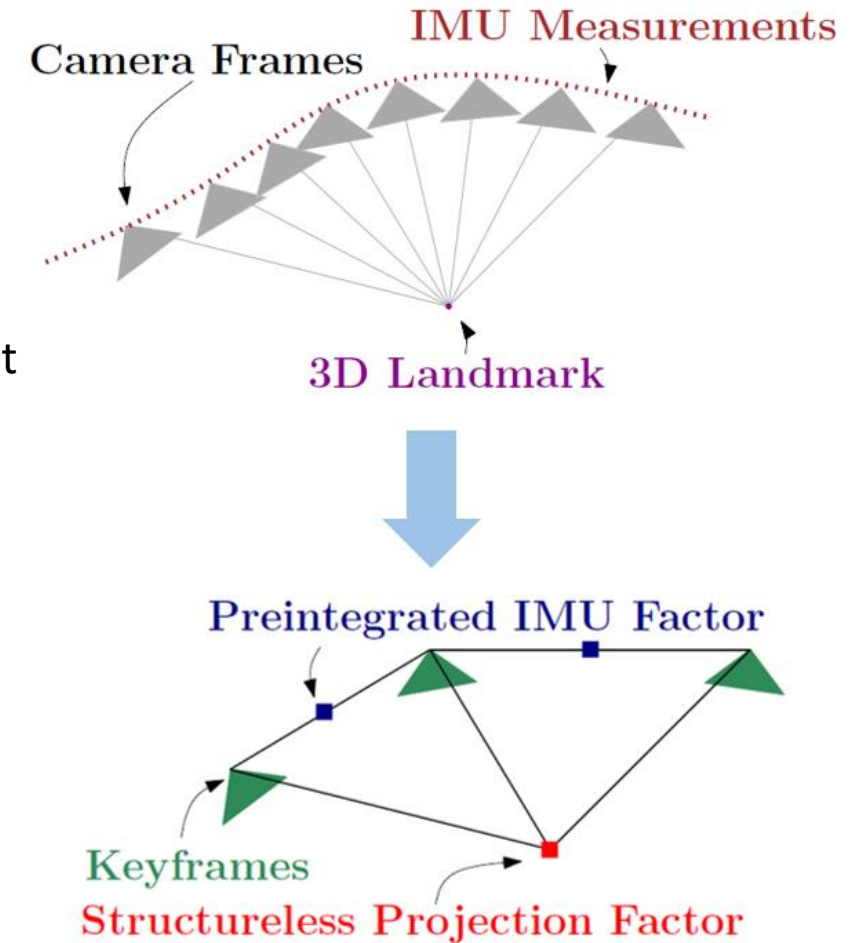


euteneier, Lynen, Bosse, Siegwart, Furgale, *Keyframe-based visual-inertial odometry using nonlinear optimization*, International Journal of Robotics Research (IJRR), 2015. [PDF](#). [Video](#). [Code](#).

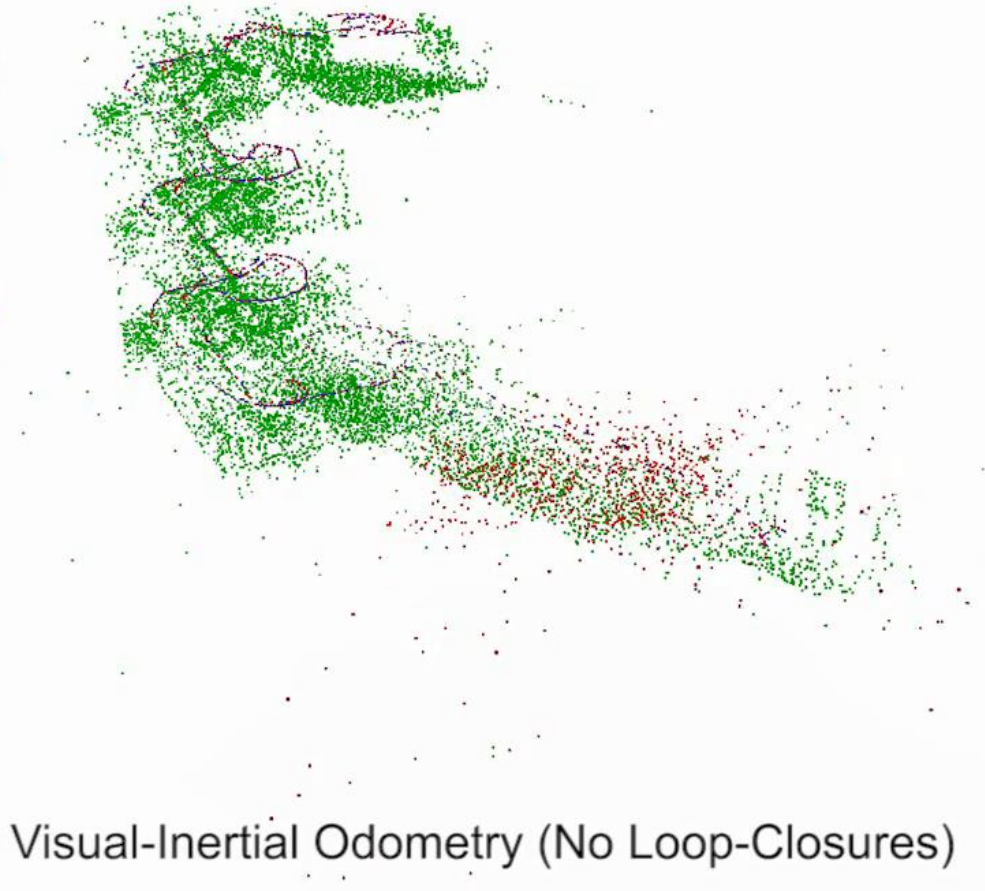
# Case Study 2: SVO+GTSAM

**Solves the same optimization problem as OKVIS but:**

- **Keeps all the frames** (from the start to the end of the trajectory)
- To make the optimization efficient
  - **Marginalizes 3D landmarks** (rather than triangulating landmarks, only their visual bearing measurements are kept and the visual constraint are enforced by the **Epipolar Line Distance** (Lecture 08))
  - **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**



# SVO+GTSAM



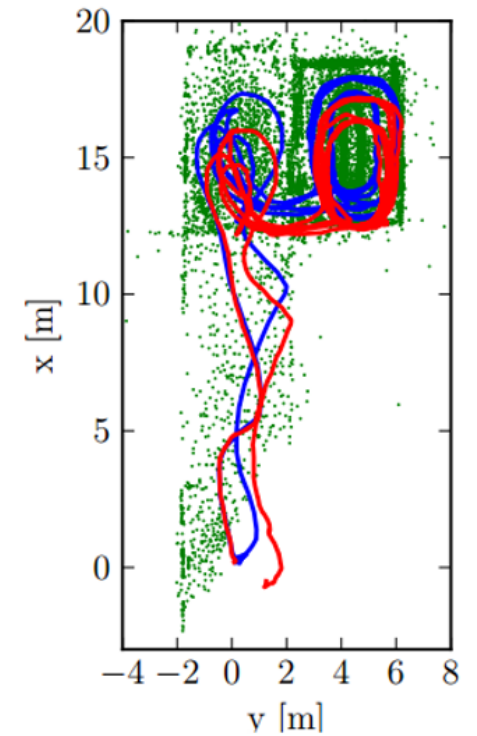
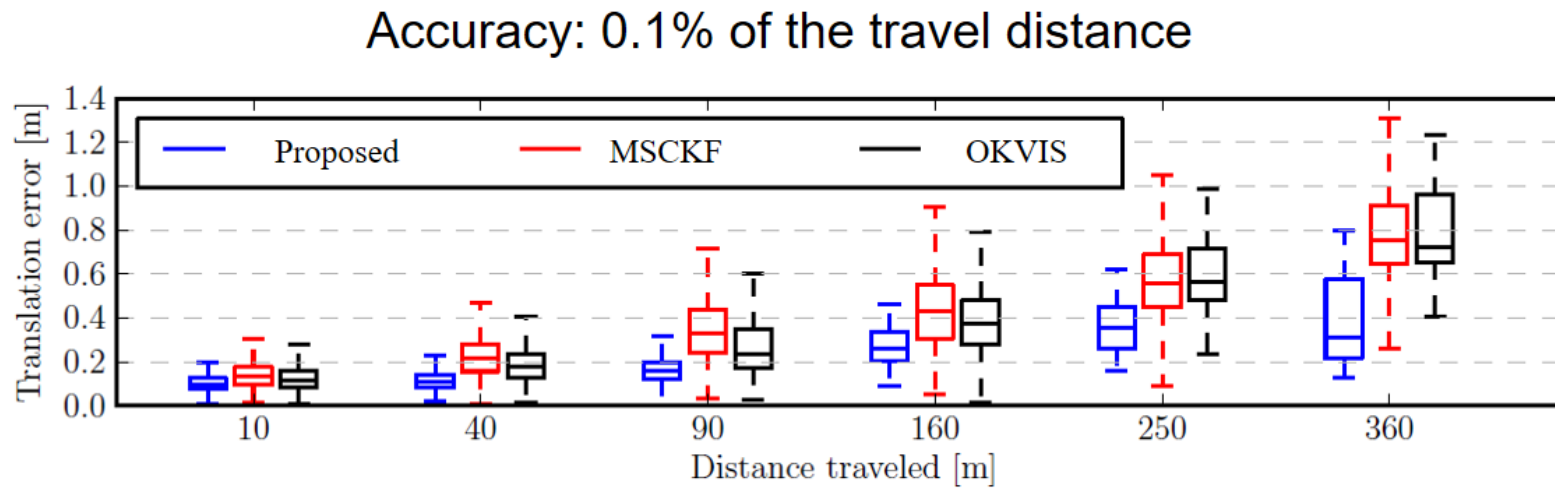
5x

Monocular Visual-Inertial Odometry (No Loop-Closures)



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



# 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

$$\{X, L, b^a, b^g\} = \underset{\{X, L, b^a, b^g\}}{\operatorname{argmin}} \left\{ \underbrace{\sum_{k=1}^N \|f(x_{k-1}, u) - x_k\|_{A_k}^2}_{\text{IMU residuals}} + \underbrace{\sum_{k=1}^N \sum_{i=1}^M \|\pi(x_k, L_i) - z_{i_k}\|_{\Sigma_{i_k}}^2}_{\substack{\text{Reprojection residuals} \\ \text{(Bundle Adjustment term)}}} \right\}$$

- **Idea: Preintegration**

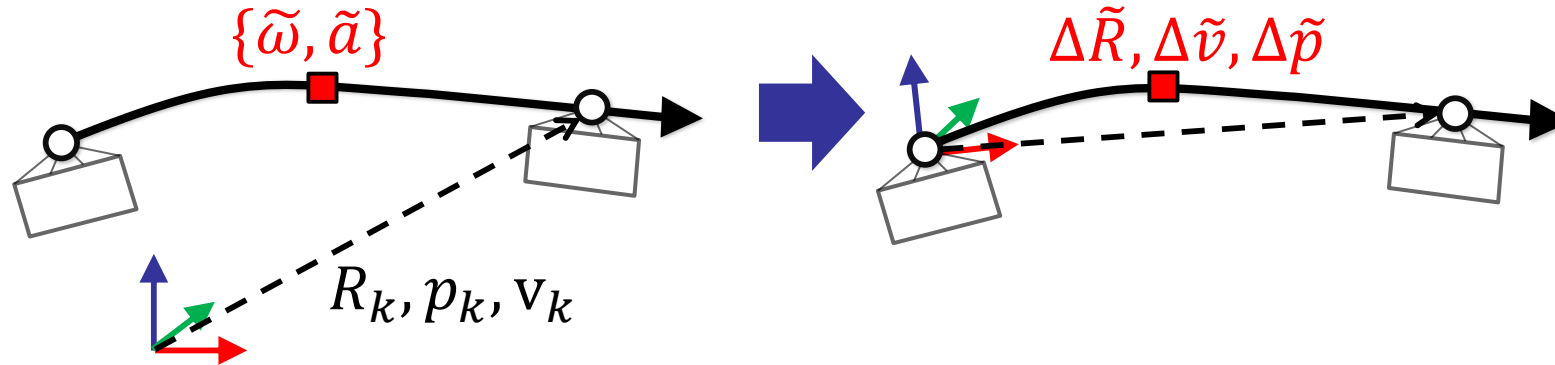
- defines relative motion increments, expressed in body frame, which are independent on the global position, orientation, and velocity at  $k$  [1]
- [2] uses this theory by leveraging the manifold structure of the rotation group  $\text{SO}(3)$

[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



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

$$\mathbf{e}_R = \hat{R}(\tilde{\omega}, R_{k-1})^T R_k$$

$$\mathbf{e}_V = \hat{v}(\tilde{\omega}, \tilde{a}, v_{k-1}) - v_k$$

$$\mathbf{e}_p = \underbrace{\hat{p}(\tilde{\omega}, \tilde{a}, p_{k-1})}_{\text{Prediction}} - \underbrace{p_k}_{\text{Estimate}}$$

Prediction Estimate

**Repeats integration** when previous state changes!

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

$$\mathbf{e}_R = \Delta\tilde{R}^T \Delta R$$

$$\mathbf{e}_V = \Delta\tilde{v} - \Delta v$$

$$\mathbf{e}_p = \Delta\tilde{p} - \Delta p$$

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

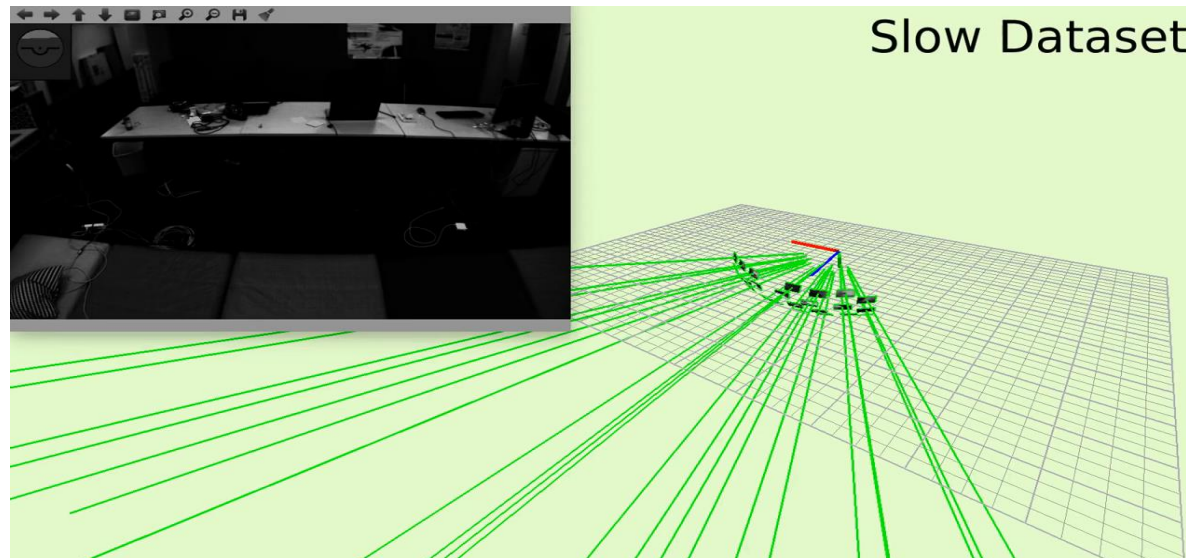


# Non-linear optimization vs. Filtering

Non-linear Optimization methods	Filtering methods
Optimize a <b>window</b> of multiple states (or all the states) using non-linear Least-Squares optimization	Solve the same problem by <b>running only one iteration of the optimization</b> function (e.g., using Extended Kalman Filter (EKF))
<ul style="list-style-type: none"><li>✓ Multiple iterations (it re-linearizes at each iteration)</li><li>✓ Achieves the highest accuracy</li><li>✓ Slower</li></ul>	<ul style="list-style-type: none"><li>✗ One iteration only</li><li>✗ Sensitive to linearization point</li><li>✓ Fastest</li></ul>

# Case Study 1: ROVIO

- EKF state:  $X = [p(t); q(t); v(t); b^a(t); b^g(t); L_1; L_2; \dots; L_k]$
- 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))



# 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**

# 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:

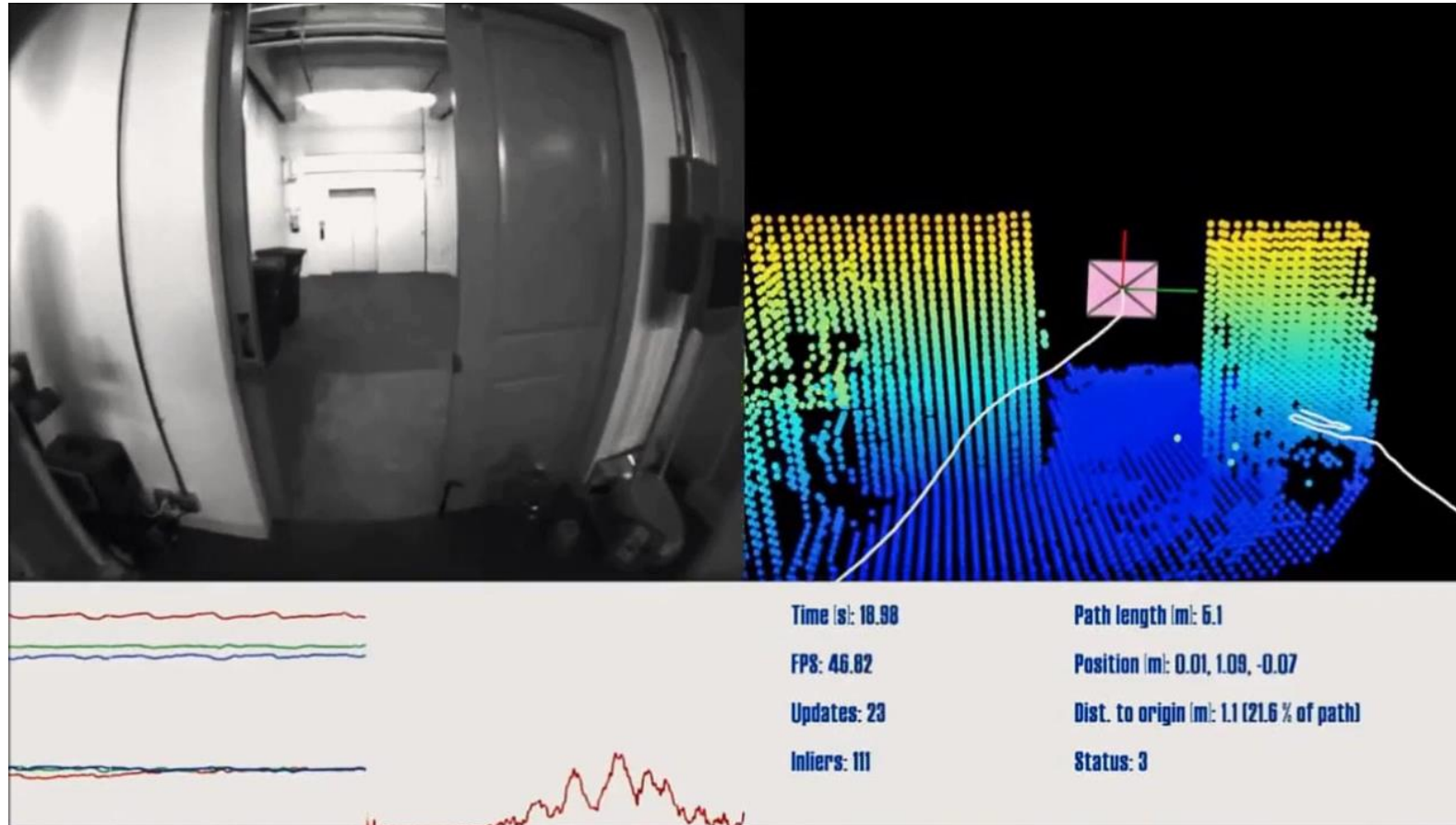
$$X = [p(t); q(t); v(t); b^a(t); b^g(t); p_{C_1}; q_{C_1}; \dots; p_{C_N}; q_{C_N}]$$

- **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**
- Released open source within the OpenVins project: [https://github.com/rpng/open\\_vins](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)



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

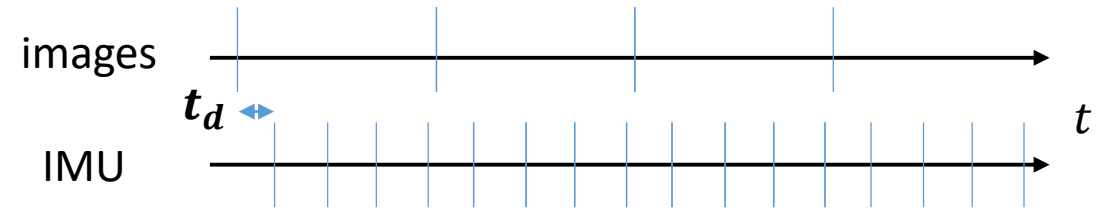
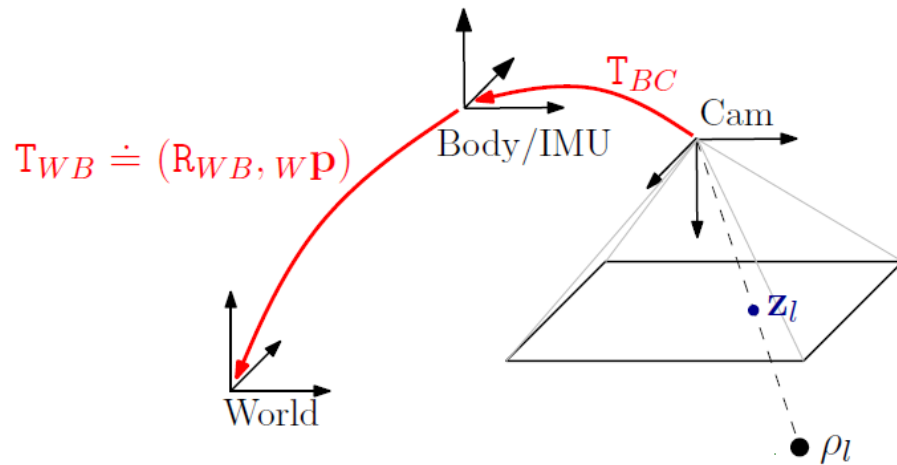
[2] Li, Mourikis, *High-precision, consistent EKF-based visual-inertial odometry*, International Journal of Robotics Research (IJRR), 2013. [PDF](#).

# 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

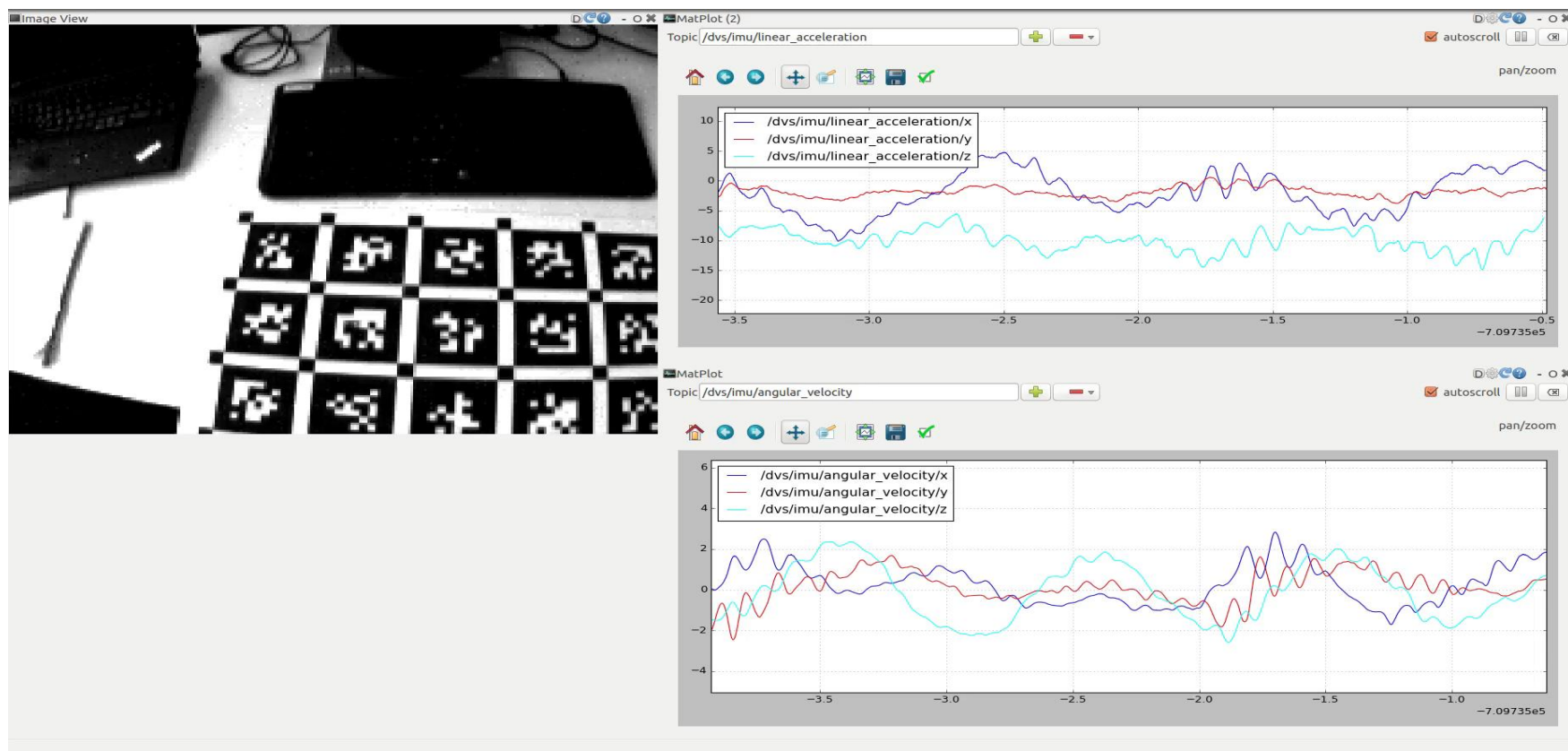
# Camera-IMU Calibration

- **Goal:** estimate the **rigid-body transformation**  $T_{BC}$  and time **offset**  $t_d$  **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>





# Kalibr Toolbox

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

$$\{X, L, T_{BC}, t_d, b^a, b^g\} = \underset{\{X, L, T_{BC}, t_d, b^a, b^g\}}{\operatorname{argmin}} \left\{ \underbrace{\sum_{k=1}^N \|f(x_{k-1}, u) - x_k\|_{\Lambda_k}^2}_{\text{IMU residuals}} + \underbrace{\sum_{k=1}^N \sum_{i=1}^M \|\pi(x_k, L_i) - z_{i_k}\|_{\Sigma_{i_k}}^2}_{\substack{\text{Reprojection residuals} \\ \text{(Bundle Adjustment term)}}} \right\}$$

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



# 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](#).

# 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?