Robotic 3D Vision

Lecture 9: Visual-Inertial Odometry

Prof. Dr. Jörg Stückler
Computer Vision Group, TU Munich
http://vision.in.tum.de

What We Will Cover Today

- Visual-Inertial Odometry (VIO)
  - Introduction
  - Inertial Measurement Units
  - Loosely vs. tightly coupled VIO

- VIO approaches
  - Filtering
  - Full-posterior optimization
  - Fixed-lag smoothing

Why Sensor Fusion for Odometry?

- Visual sensors have limitations
  - Rapid motion causes motion blur: less accuracy/robustness
  - Degenerate motion/reconstruction in textureless areas
  - Illumination conditions
  - Limited frame-rate (30-60Hz for typical cameras)
  - Monocular vision: scale ambiguity

- Idea: complement visual motion estimation with other sensing modalities

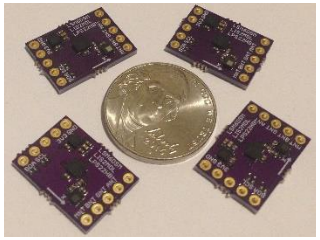
Inertial Measurement Units

- Inertial Measurement Units (IMUs) measure 3-axis linear accelerations and angular velocities of the sensor wrt the earth (inertial) reference frame

Mechanical IMU of Saturn rocket

- Mechanical, optical and micro-electro-mechanical systems (MEMS) sensors

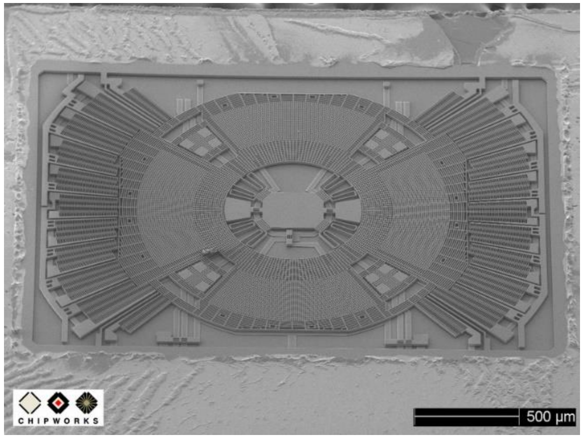
- Robotics:
  - MEMS sensors are cheap, small, lightweight, power-efficient, solid state and often sufficiently accurate/robust

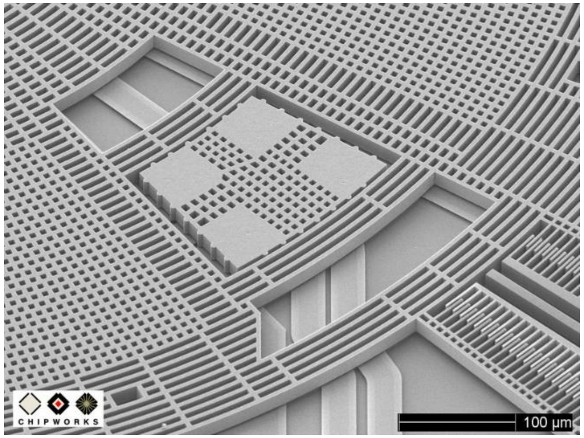


Gyroscopes

- Historically, first gyroscopes have been mechanical devices
- Today also MEMS technology
  - Measure displacement by Coriolis force of rotary vibrating structure through capacitive electrodes

Mechanical IMU of Saturn rocket

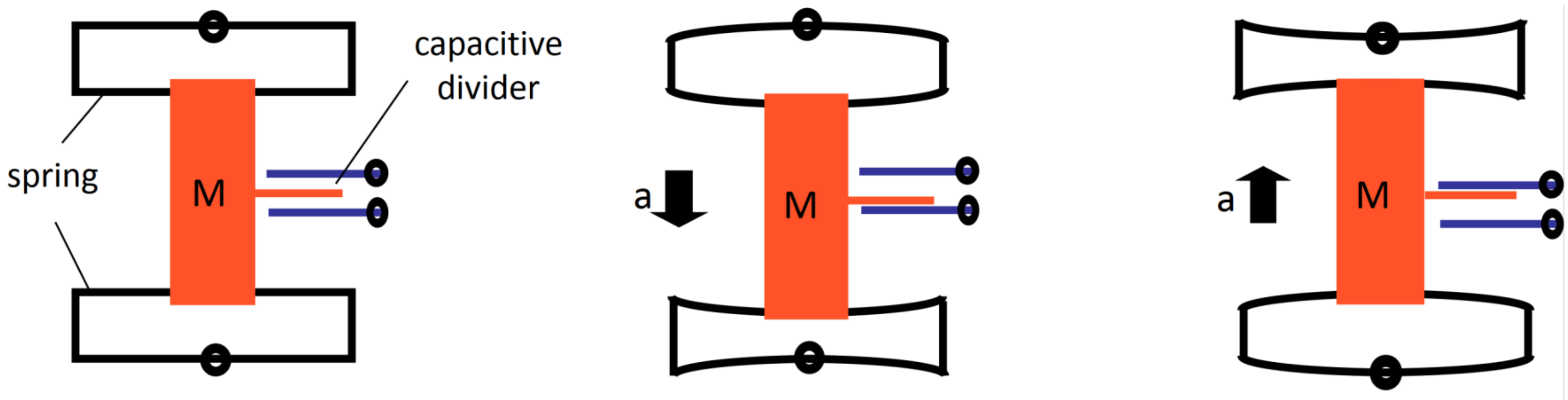




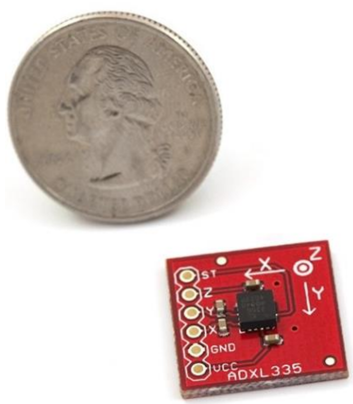
Fiber Optic Gyroscope

MEMS Gyroscope (ST LYPR540AH)

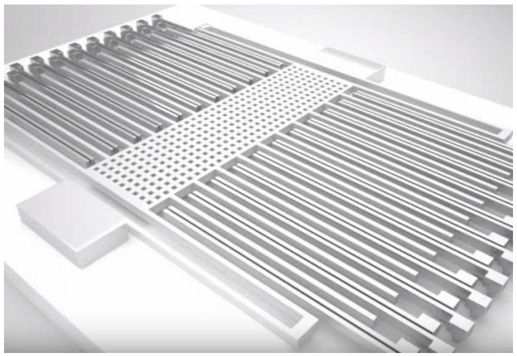
Accelerometers



- MEMS accelerometers measure displacement of spring mounted fin structure using capacitive electrodes
- Fin structure moves due to accelerations



3-axis ADXL 335



MEMS accelerometer

IMU Odometry

- Idea: integrate rotational velocities and double-integrate accelerations to estimate position
- This will lead to large drift, especially for cheap IMUs!
  - Intuition (math later):
    - Integration of angular velocity: orientation error proportional in t
    - Double-integration of linear accelerations: position error proportional in t^2
    - Gyro and accelerometer measurements are biased, additional error
    - Orientation errors cause further position error (why?)

	Accelerometer Bias Error		Horizontal Position Error [m]			
Grade	[mg]		1s	10s	60s	1hr
Navigation	0.025		0.13 mm	12 mm	0.44 m	1.6 km
Tactical	0.3		1.5 mm	150 mm	5.3 m	19 km
Industrial	3		15 mm	1.5 m	53 m	190 km
Automotive	125		620 mm	60 m	2.2 km	7900 km

http://www.vectornav.com/support/library/imu-and-ins

Visual-Inertial Fusion

- Vision and IMU are complementary!

Visual sensing	Inertial sensing
+ Accurate at small to medium motion	- Large relative uncertainty for low acceleration/angular velocity
+ Rich information for other purposes	
- Limited output rate (~100Hz)	+ High output rate (~1000Hz)
- Scale ambiguity for monocular camera	+ Scale directly observable
- Lack of robustness for rapid motion, textureless areas, low illumination	+ Independent of environmental conditions

- Odometry using both sensor types is still prone to drift!

IMU Measurement Model

- IMU measures angular velocity $\boldsymbol{\omega}$ and linear acceleration $\mathbf{a}$ in "body" frame:

$$_B\tilde{\boldsymbol{\omega}}_B^W(t) = _B\boldsymbol{\omega}_B^W(t) + \mathbf{b}_{\boldsymbol{\omega}}(t) + \boldsymbol{\epsilon}_{\boldsymbol{\omega}}(t)$$

$$_B\tilde{\mathbf{a}}_B^W = \mathbf{R}_W^B(t)\left(_W\mathbf{a}_B^W(t) - _W\mathbf{g}\right) + \mathbf{b}_{\mathbf{a}}(t) + \boldsymbol{\epsilon}_{\mathbf{a}}(t)$$

- B: body frame, W: world frame
- Left subscript X: quantity expressed in frame X
- Right subscript X, superscript Y: quantity of frame X wrt frame Y
- Bias terms $\mathbf{b}_{\boldsymbol{\omega}}(t)$ and $\mathbf{b}_{\mathbf{a}}(t)$
- Noise terms $\boldsymbol{\epsilon}_{\boldsymbol{\omega}}(t)$ and $\boldsymbol{\epsilon}_{\mathbf{a}}(t)$

IMU Noise Model

- Noise terms $\boldsymbol{\epsilon}_{\boldsymbol{\omega}}(t)$ and $\boldsymbol{\epsilon}_{\mathbf{a}}(t)$
  - Zero-mean Gaussian noise

- Bias terms $\mathbf{b}_{\boldsymbol{\omega}}(t)$ and $\mathbf{b}_{\mathbf{a}}(t)$
  - Bias drifts due to temperature change, pressure change, etc.
  - Random walk (derivative is Gaussian white noise)

$$\dot{\mathbf{b}}_{\boldsymbol{\omega}}(t) = \delta_{\boldsymbol{\omega}}(t) \qquad \delta_{\boldsymbol{\omega}}(t) \sim \mathcal{N}(0, \Delta t \sigma_{\boldsymbol{\omega}}^2)$$

$$\dot{\mathbf{b}}_{\mathbf{a}}(t) = \delta_{\mathbf{a}}(t) \qquad \delta_{\mathbf{a}}(t) \sim \mathcal{N}(0, \Delta t \sigma_{\mathbf{a}}^2)$$

- Bias needs to be estimated as well!

IMU Integration

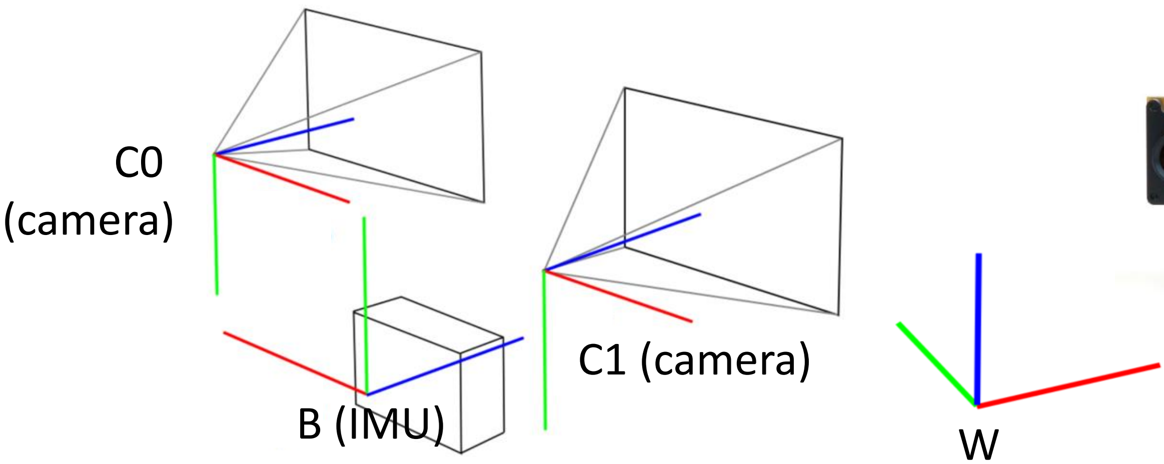
- Per-component integration
- Integrate gyro measurements to obtain rotation
- Double-integrate accelerations based on rotation estimate

$$\mathbf{p}(t_2) = \mathbf{p}(t_1) + \mathbf{v}(t_1)(t_2 - t_1) + \int \int_{t_1}^{t_2} \mathbf{R}(t) \left(\widetilde{\mathbf{a}}(t) - \mathbf{b}_{\mathbf{a}}(t) \right) + \mathbf{g} \, dt^2$$

- Problems:
  - Requires known biases
  - Requires known initial rotation
  - Does not make use of known gravity direction for rotation estimate

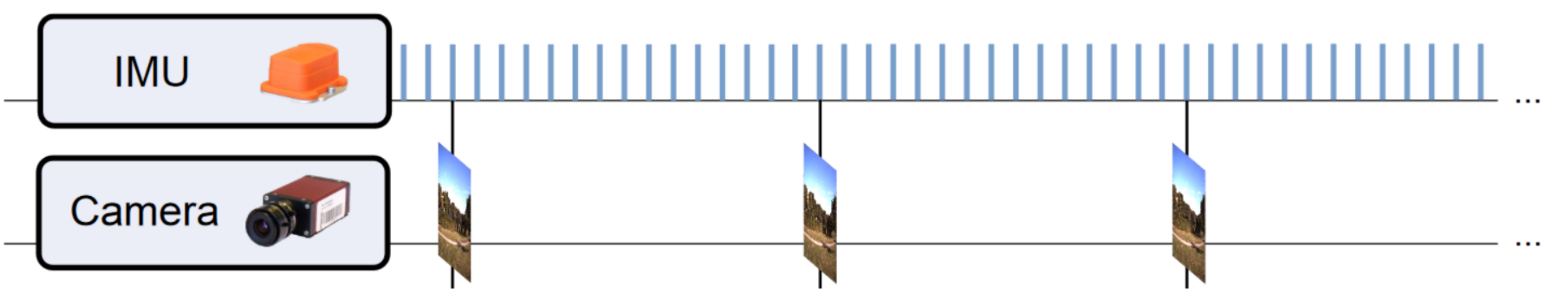
Camera-IMU System

- Extrinsic calibration between camera(s) and IMU frame



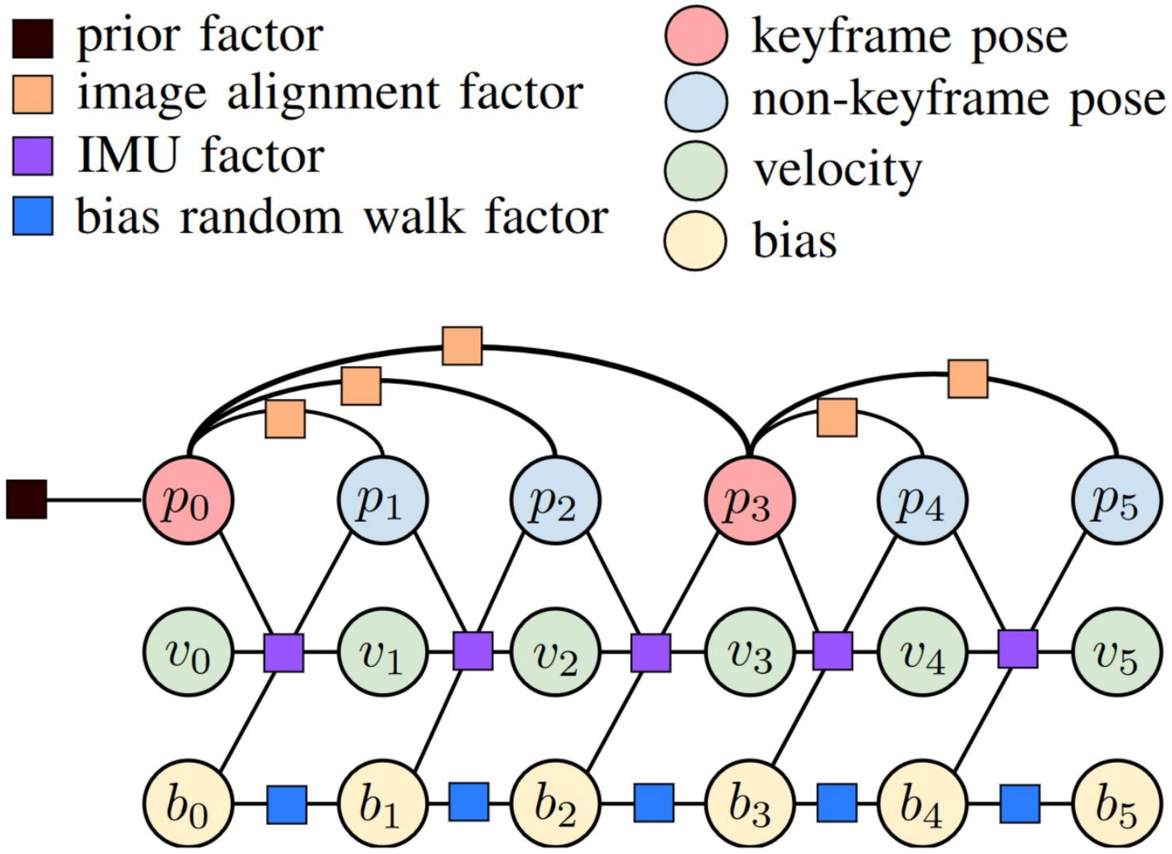
Skybotix VI-Sensor

- Time synchronization



Factor Graph of Visual-Inertial Odometry

(Keyframe-based)



Loosely vs. Tightly Coupled Fusion

- Different paradigms to fuse visual and inertial measurements
- Loosely coupled: fuse estimates of individual sensors
  - First estimate pose independently from each sensor type
  - Fuse the pose estimates
- Tightly coupled: estimate state directly from measurements of both sensors
  - Fuse raw measurements, i.e. IMU measurements, keypoints, direct image alignment, etc.
  - Examples:
    - Combined error function of reprojection and IMU residuals
    - Prediction with IMU to confine image search regions for keypoint matching
  - More accurate but higher implementation effort

State Estimation Approaches

Filtering	Fixed-Lag Smoothing	Maximum-A-Posteriori (MAP) Estimation
Recursive Bayesian filtering of the most recent state (e.g. Kalman Filter)	Optimize window of states through non-linear optimization and marginalization of old states	Full posterior optimization of all states through non-linear least squares
- Single linearization	+ Relinearize (in window)	+ Relinearize
- Accumulation of linearization errors	- Accumulation of linearization errors	+ Sparse Matrices
- Gaussian approximation of marginalized states	- Gaussian approximation of marginalized states	+ Highest Accuracy
+ Faster	+ Fast	+ Slow

Recap: Extended Kalman Filter (EKF)

- Non-linear state-transition model with Gaussian noise:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{d_t})$$

- Non-linear observation model with Gaussian noise:

$$\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{\delta}_t \qquad \boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{m_t})$$

- How to cope with non-linear system?

- Idea: linearize the models in each time step

$$\Rightarrow \quad \mathbf{x}_t \approx g(\mathbf{x}_{t-1}^0, \mathbf{u}_t) + \nabla g(\mathbf{x}, \mathbf{u}_t)\big|_{\mathbf{x}=\mathbf{x}_{t-1}^0} \left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^0\right) + \boldsymbol{\epsilon}_t$$

$$\Rightarrow \quad \mathbf{y}_t \approx h(\mathbf{x}_t^0) + \nabla h(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_t^0} \left(\mathbf{x}_t - \mathbf{x}_t^0\right) + \boldsymbol{\delta}_t$$

Recap: EKF Prediction & Correction

- Efficient approximate correction and prediction steps which involve manipulation of Gaussians and linearization
- The state estimate can be represented as a Gaussian distribution

$$\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$$

- Prediction: $\boldsymbol{\mu}_t^- = g(\boldsymbol{\mu}_{t-1}^+, \mathbf{u}_t)$

$$\boldsymbol{\Sigma}_t^- = \mathbf{G}_t \boldsymbol{\Sigma}_{t-1}^+ \mathbf{G}_t^\top + \boldsymbol{\Sigma}_{d_t} \qquad \mathbf{G}_t := \nabla g(\mathbf{x}, \mathbf{u}_t)\big|_{\mathbf{x}=\boldsymbol{\mu}_{t-1}^+}$$

- Correction: $\mathbf{K}_t = \boldsymbol{\Sigma}_t^- \mathbf{H}_t^\top \left(\mathbf{H}_t \boldsymbol{\Sigma}_t^- \mathbf{H}_t^\top + \boldsymbol{\Sigma}_{m_t}\right)^{-1}$

$$\boldsymbol{\mu}_t^+ = \boldsymbol{\mu}_t^- + \mathbf{K}_t \left(\mathbf{y}_t - h(\boldsymbol{\mu}_t^-)\right) \qquad \mathbf{H}_t := \nabla h(\mathbf{x})\big|_{\mathbf{x}=\boldsymbol{\mu}_t^-}$$

$$\boldsymbol{\Sigma}_t^+ = \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t\right) \boldsymbol{\Sigma}_t^-$$

Tightly-Coupled Filter for Visual-Inertial Fusion

- State-transition model: IMU model integration + noise
- Use IMU model as state-transition model

$$
\begin{aligned}
{}_B\widetilde{\boldsymbol{\omega}}_B^W(t) &= {}_B\boldsymbol{\omega}_B^W(t) + \mathbf{b}_{\boldsymbol{\omega}}(t) + \boldsymbol{\epsilon}_{\boldsymbol{\omega}}(t) \\
{}_B\widetilde{\mathbf{a}}_B^W(t) &= \mathbf{R}_W^B \left({}_W\mathbf{a}_B^W(t) - {}_W\mathbf{g}\right) + \mathbf{b}_{\mathbf{a}}(t) + \boldsymbol{\epsilon}_{\mathbf{a}}(t) \\
\dot{\mathbf{b}}_{\boldsymbol{\omega}}(t) &= \delta_{\boldsymbol{\omega}}(t) \\
\dot{\mathbf{b}}_{\mathbf{a}}(t) &= \delta_{\mathbf{a}}(t)
\end{aligned}
$$

- Integrate measurements to propagate state

$$
\mathbf{p}(t_2) = \mathbf{p}(t_1) + \mathbf{v}(t_1)(t_2 - t_1) + \int\int_{t_1}^{t_2} \mathbf{R}(t)\left(\widetilde{\mathbf{a}}(t) - \mathbf{b}_{\mathbf{a}}(t)\right) - \mathbf{g}\, dt^2
$$

- Approximate Gaussian noise in propagated state

Tighly-Coupled Filter for Visual-Inertial Fusion

- Measurement model: visual measurements
  - Example: keypoint reprojections

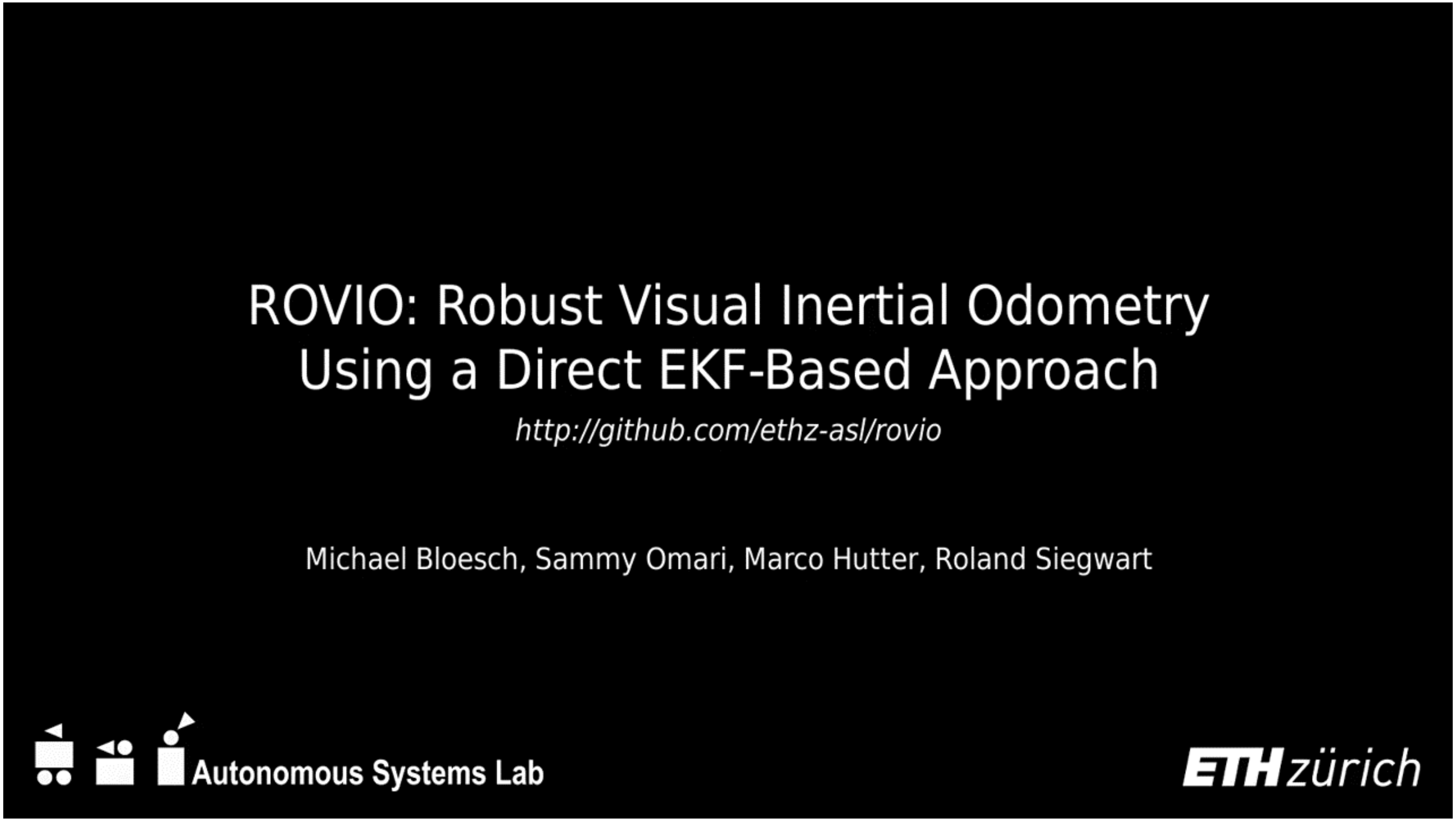
$$\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{\delta}_t \qquad \boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{m_t})$$

  - Requires 3D landmark positions in state

$$\mathbf{x}_t = (\mathbf{p}_t, \mathbf{R}_t, \mathbf{v}_t, \boldsymbol{\omega}_t, \mathbf{l}_{t,1}, \ldots, \mathbf{l}_{t,N})$$

Tightly-Coupled Filter for Visual-Inertial Fusion

- Photoconsistency measurements of landmark patch projections



Drawbacks of Filter-based Approaches

- Linearization errors
  - Linearization with the current state estimate introduces linearization erros
  - Marginalization of old states, no reoptimization or relinearization possible
  - Leads to inconsistency of the mean/covariance estimate
- Wrong covariances/initial states
  - Modeled noise in measurement and state-transition may be inaccurate
  - Leads to over- or underconfident estimates
- Number of visual landmarks needs to be limited due to quadratic run-time in state variable dimension

Multi-State Constraint Kalman Filter

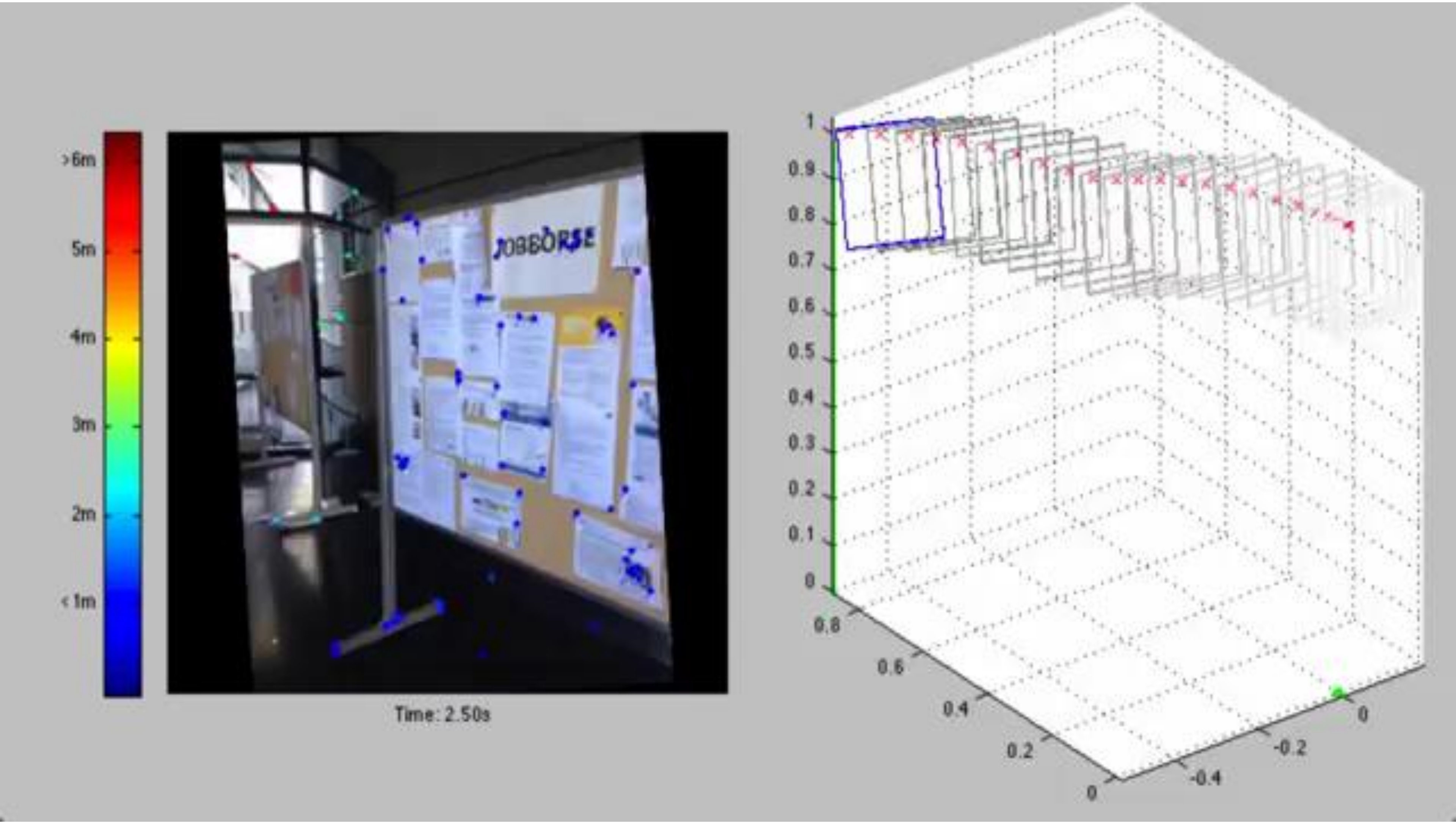
- Mourikis and Roumeliotis, A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation, ICRA 2007

- Visual-inertial filter without landmarks in the state
  - Include a window of current positions, velocities, biases and rotations into state

  - Perform local least-squares estimate to reconstruct keypoint matches observed in multiple cameras of the optimization window

  - Add measurement residuals for each camera pose and reconstructed keypoints (needs decorrelation of pose errors and feature position errors in residuals)

Multi-State Constraint Kalman Filter



M. Shelley, MSc thesis, TUM

Recap: Full State Posterior Factorization

- The full state posterior factorizes into a product of observation likelihoods, state-transition likelihoods and the initial state distribution

$$
\begin{aligned}
p\left(X_{0:t} \middle| U_{1:t}, Y_{0:t}\right) &= \frac{p\left(Y_t \middle| X_{0:t}, U_{1:t}, Y_{0:t-1}\right) p\left(X_{0:t} \middle| U_{1:t}, Y_{0:t-1}\right)}{p\left(Y_t \middle| U_{1:t}, Y_{0:t}\right)} \\
&= \frac{p\left(Y_t \middle| X_t\right) p\left(X_t \middle| X_{0:t-1}, U_{1:t}, Y_{0:t-1}\right) p\left(X_{0:t-1} \middle| U_{1:t}, Y_{0:t-1}\right)}{p\left(Y_t \middle| U_{1:t}, Y_{0:t}\right)} \\
&= \eta_t \, p\left(Y_t \middle| X_t\right) p\left(X_t \middle| X_{t-1}, U_t\right) p\left(X_{0:t-1} \middle| U_{1:t-1}, Y_{0:t-1}\right) \\
&= p\left(X_0\right) \left(\prod_{\tau=0}^{t} \eta_\tau \, p\left(Y_\tau \middle| X_\tau\right) \right) \left(\prod_{\tau=1}^{t} p\left(X_\tau \middle| X_{\tau-1}, U_\tau\right) \right)
\end{aligned}
$$

Recap: Full State Posterior – Non-Linear Gaussian Case

$$p\left(X_{0:t} \mid U_{1:t}, Y_{0:t}\right) = p\left(X_0\right)\left(\prod_{\tau=0}^{t} \eta_\tau \, p\left(Y_\tau \mid X_\tau\right)\right)\left(\prod_{\tau=1}^{t} p\left(X_\tau \mid X_{\tau-1}, U_\tau\right)\right)$$

- Non-linear state-transition model with Gaussian noise:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) + \boldsymbol{\epsilon}_t$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{d_t})$$

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_t; g(\mathbf{x}_{t-1}, \mathbf{u}_t), \boldsymbol{\Sigma}_{d_t})$$

- Non-linear observation model with Gaussian noise:

$$\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{\delta}_t$$

$$\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{m_t})$$

$$p(\mathbf{y}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t; h(\mathbf{x}_t), \boldsymbol{\Sigma}_{m_t})$$

- Gaussian initial state estimate:

$$\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \qquad p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

Recap: Non-Linear Least Squares

- We can rewrite the negative log-posterior as a non-linear least squares problem:

$$\arg\min_{\mathbf{x}} E(\mathbf{x}) = \frac{1}{2}\mathbf{r}(\mathbf{x})^\top \mathbf{W}\mathbf{r}(\mathbf{x})$$

  - Stack residuals in residual vector $\mathbf{r}(\mathbf{x})$
  - Inverse covariances in block-diagonal weight matrix $\mathbf{W}$

- Optimization approaches:
  - Gradient descent
  - Gauss-Newton
  - Levenberg-Marquardt
  - etc.

Recap: Gauss-Newton Method

- Idea: Approximate Newton's method to minimize E(x)
  - Approximate E(x) through linearization of residuals

$$
\begin{aligned}
\widetilde{E}(\mathbf{x}) &= \frac{1}{2}\widetilde{\mathbf{r}}(\mathbf{x})^\top \mathbf{W} \widetilde{\mathbf{r}}(\mathbf{x}) \\
&= \frac{1}{2}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x} - \mathbf{x}_k\right)\right)^\top \mathbf{W}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x} - \mathbf{x}_k\right)\right) \qquad \mathbf{J}_k := \nabla_{\mathbf{x}} \mathbf{r}(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_k} \\
&= \frac{1}{2}\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W} \mathbf{r}(\mathbf{x}_k) + \underbrace{\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W} \mathbf{J}_k}_{=:\mathbf{b}_k^\top}\left(\mathbf{x} - \mathbf{x}_k\right) + \frac{1}{2}\left(\mathbf{x} - \mathbf{x}_k\right)^\top \underbrace{\mathbf{J}_k^\top \mathbf{W} \mathbf{J}_k}_{=:\mathbf{H}_k}\left(\mathbf{x} - \mathbf{x}_k\right)
\end{aligned}
$$

  - Find root of $\nabla_{\mathbf{x}} \widetilde{E}(\mathbf{x}) = \mathbf{b}_k^\top + \left(\mathbf{x} - \mathbf{x}_k\right)^\top \mathbf{H}_k$ using Newton's method, i.e.

$$
\nabla_{\mathbf{x}} \widetilde{E}(\mathbf{x}) = \mathbf{0} \text{ iff } \mathbf{x} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{b}_k
$$

- Pros:
  - Faster convergence (approx. quadratic convergence rate)
- Cons:
  - Divergence if too far from local optimum (H not positive definite)
  - Solution quality depends on initial guess

Full State Posterior Optimization

- Idea: use state-transition and measurement models to formulate non-linear least squares optimization problem to obtain a full state posterior estimate

- Requires iterative reintegration and relinearization
  - Relinearize in each iteration in current state estimate
    - Slow but more accurate than filter
  - Evaluation of the state-transition residuals requires integration of the state-variables at time t+1 from IMU measurements and state estimate at time t

$$p(t_2) = p(t_1) + v(t_1)(t_2 - t_1) + \int \int_{t_1}^{t_2} \mathbf{R}(\boldsymbol{\omega}(t)) \left(\widetilde{\mathbf{a}}(t) - \mathbf{b}_{\mathbf{a}}(t) \right) + \mathbf{g} \, dt^2$$

  - Optimization problem quickly becomes large (and slow)

IMU Preintegration

- State-transitions need to be reintegrated in each time step due to its dependency of the formulation on the start state
  - Start state is expressed in the world frame
  - Start state changes due to optimization
- Integrate "relative motion" $\Delta \boldsymbol{p}_{i \to j}$, $\Delta \boldsymbol{v}_{i \to j}$, and $\mathbf{R}_{i \to j}$ between frames, starting at zero motion:

$$\begin{aligned} \Delta \boldsymbol{p}_{i \to k+1} &= \Delta \boldsymbol{p}_{i \to k} + \Delta \boldsymbol{v}_{i \to k} \Delta t \\ \Delta \boldsymbol{v}_{i \to k+1} &= \Delta \boldsymbol{v}_{i \to k} + \mathbf{R}_{i \to k} \left(\boldsymbol{a}_z - \boldsymbol{b}_a \right) \Delta t \\ \mathbf{R}_{i \to k+1} &= \mathbf{R}_{i \to k} \exp\left(\left[\boldsymbol{\omega}_z - \boldsymbol{b}_\omega \right]_\times \Delta t \right) \end{aligned}$$

- Compare with relative motion between estimates
- Neglecting small changes in bias, "preintegration" becomes possible

Fixed-Lag Smoothing

- Still, optimizing the full state posterior is too slow for real-time odometry

- Can we…
  - … efficiently optimize a window of frames as in non-linear least squares?
  - … marginalize old state variables as in filtering?

- Let's look closer at the Gauss-Newton update step…

Fixed-Lag Smoothing

- Let's look closer at the Gauss-Newton update step…

$$\nabla_{\mathbf{x}} \widetilde{E}(\mathbf{x}) = \mathbf{0} \text{ iff } \mathbf{x} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{b}_k$$

- We need to invert the Hessian H to solve for the update on x
- Can we reduce the system of linear equations to update only a small set of state variables in a recent optimization window?
- Idea: Apply the Schur complement (corresponds to marginalization of the Gaussian full state estimate)

Marginalization of Old States

- Let's look closer at the Gauss-Newton update step…

$$\nabla_{\mathbf{x}} \widetilde{E}(\mathbf{x}) = \mathbf{0} \text{ iff } \mathbf{x} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{b}_k$$

- We need to invert the Hessian H to solve for the update on x

$$\mathbf{H} \Delta \mathbf{x} = -\mathbf{b} \qquad \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{x}_1 \\ \Delta \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} -\mathbf{b}_1 \\ -\mathbf{b}_2 \end{pmatrix}$$

- We can solve for $\Delta \mathbf{x}_1$ using the Schur complement of $\mathbf{H}_{22}$ in H:

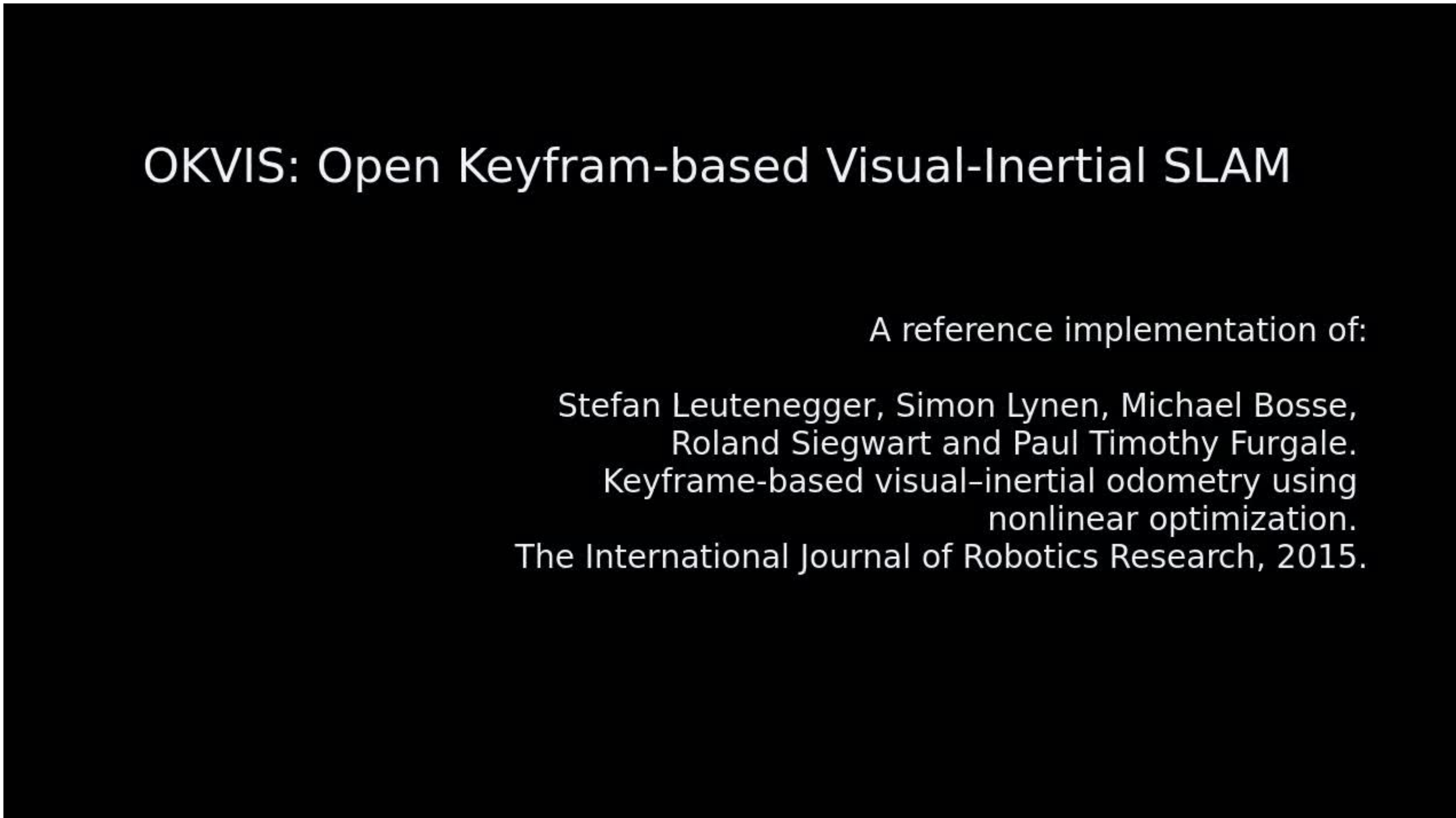
$$\left(\mathbf{H}_{11} - \mathbf{H}_{12} \mathbf{H}_{22}^{-1} \mathbf{H}_{21} \right) \Delta \mathbf{x}_1 = -\mathbf{b}_1 + \mathbf{H}_{12} \mathbf{H}_{22}^{-1} \mathbf{b}_2$$

Pros and Cons of Fixed-Lag Smoothing

- Marginalization leads to fill-in of the Hessian (will see later why it is sparse initially)
  - Inversion (of submatrices) gets more costly
- Dropping of states (e.g. landmarks) avoids fill-in but reduces accuracy
  - Trade-off between accuracy and efficiency
- Marginalization fixes the linearization point of the marginalized state variables
  - Linearization errors cannot be corrected
  - Variables may be linearized multiple times at different values
  - Estimate becomes inconsistent/inaccurate

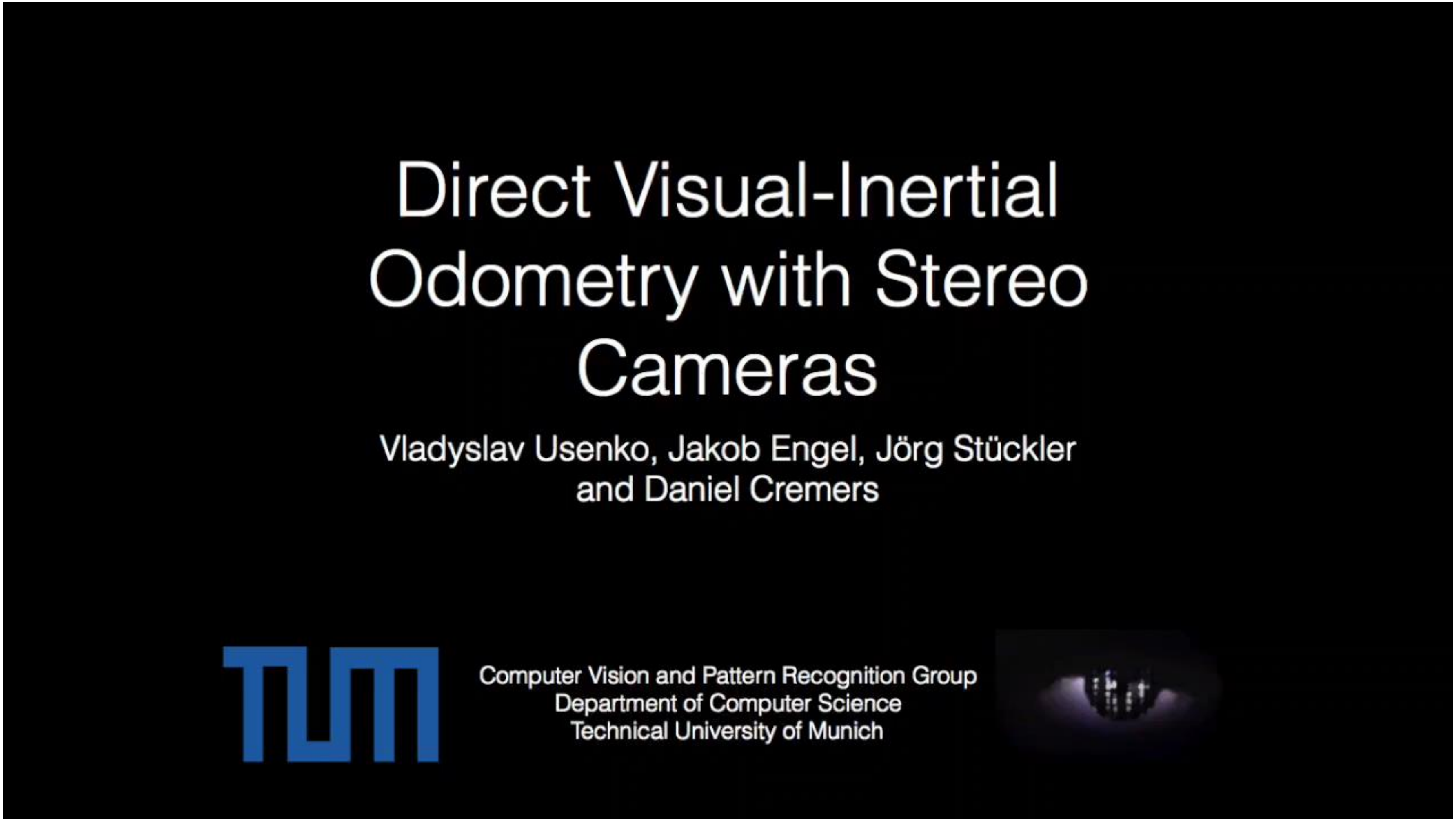
Indirect Fixed-Lag Smoothing Example

- OKVIS: Keyframe-based indirect fixed-lag smoothing VIO

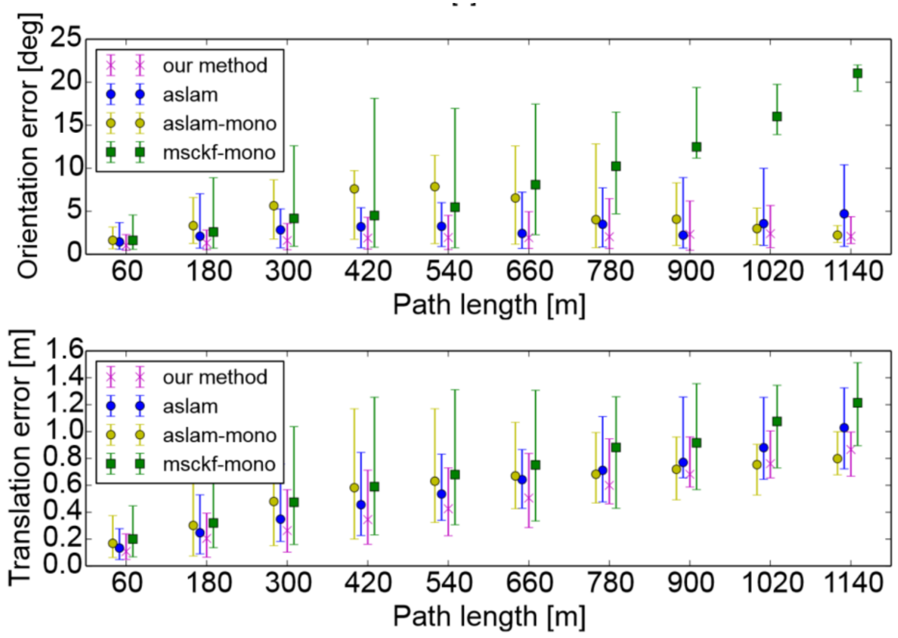


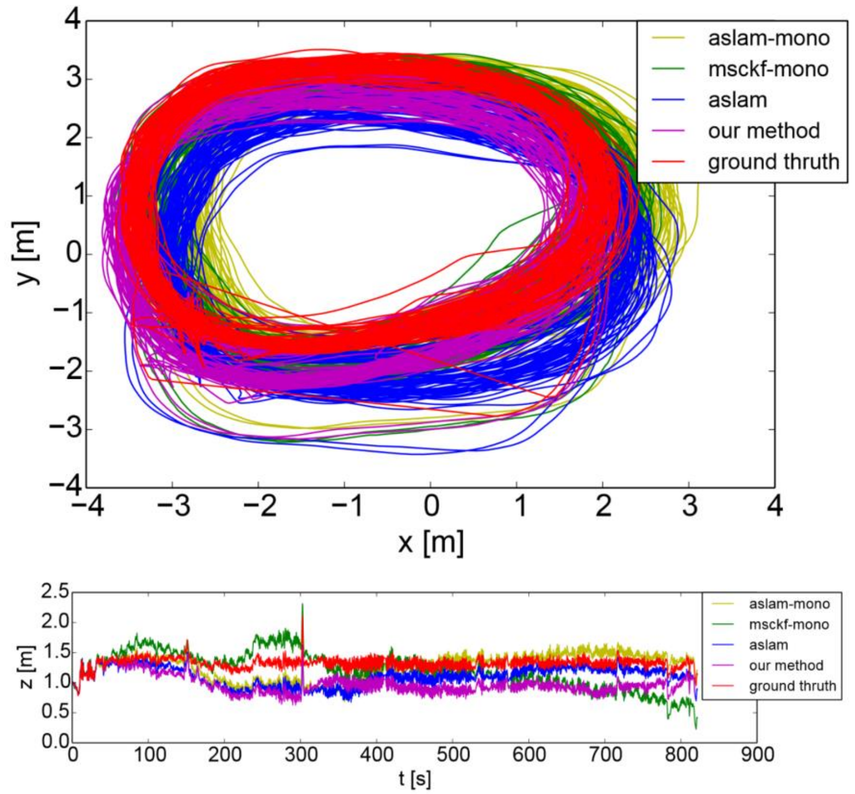
Direct Fixed-Lag Smoothing Example

- Direct Fixed-Lag Smoothing VIO



Fixed-Lag Smoothing Example Results





Lessons Learned Today

- Vision and inertial sensors complement each others well for accurate and robust visual-inertial odometry (VIO)
- Loosely-coupled vs. tightly coupled VIO
- Filtering-based methods are fast but can only optimize the state variables of the most recent time step
- Full state posterior optimization too slow for real-time performance, but most accurate
- Fixed-lag smoothing as a trade-off:
  - optimization of a window of recent time steps
  - marginalization of old states

Further Reading

- Visual-Inertial Odometry
  - Filtering:
    - Bloesch et al., Robust Visual Inertial Odometry Using a Direct EKF-Based Approach, IROS 2015
    - Mourikis and Roumeliotis, A multi-state constraint Kalman filter for vision-aided inertial navigation, ICRA 2007
  - Fixed-lag smoothing:
    - Leutenegger et al., Keyframe-based visual–inertial odometry using nonlinear optimization, IJRR 2014
    - Usenko et al., Direct Visual-Inertial Odometry with Stereo Cameras, ICRA 2016
- IMU Preintegration
  - Forster et al., IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation, RSS 2015

Thanks for your attention!