Visual Inertial Navigation
Short Tutorial

Stergios Roumeliotis
University of Minnesota
Outline

• VINS Introduction
• IMU/Camera: Models, spatial/temporal calibration
• Image Processing: Feature extraction, tracking, loop closure detection
• VIO/SLAM
  • MSCKF feature classification/processing
  • MSCKF and its (mysterious) relation to optimization methods
  • Observability and inconsistency
• Mapping
  • Offline/online, centralized/distributed approaches
  • Map-based updates and inconsistency
• Interesting Research Directions

Introduction

• Visual Inertial Navigation Systems (VINS) combine camera and IMU measurements in real time to
  • Determine 6 DOF position & orientation (pose)
  • Create 3D map of surroundings

• Applications
  • Autonomous navigation, augmented/virtual reality

• VINS advantage: IMU-camera complementary sensors -> low cost/high accuracy
IMU Model

- IMU Measurement Model
  - Gyroscope: \( \omega_m(t) = I\omega(t) + b_g(t) + n_g(t) \)
  - Accelerometer: \( a_m(t) = C(Iq_G(t))(G a(t) - G g) + b_a(t) + n_a(t) \)

- Continuous-time System Equations

\[
\begin{align*}
\dot{I}q_G(t) &= \frac{1}{2} \Omega(\omega_m(t) - b_g(t) - n_g(t))Iq_G(t) \\
\dot{b}_g(t) &= \mathbf{n}_{wg}(t) \\
G\dot{v}_I(t) &= C^T(Iq_G(t))(a_m(t) - b_a(t) - n_a(t)) + Gg \\
\dot{b}_a(t) &= \mathbf{n}_{wa}(t) \\
G\dot{p}_I(t) &= Gv_I(t)
\end{align*}
\]

\[
\begin{align*}
\dot{x}_I(t) &= f_c(x_I(t), u(t) - n(t)) \\
x_I &= [Iq_G^T \quad b_g^T \quad Gv_I^T \quad b_a^T \quad Gp_I^T]^T \\
u(t) &= [\omega_m(t)^T \quad a_m(t)^T]^T
\end{align*}
\]

- \( q \): Quaternion of orientation
- \( C \): Rotation matrix
- \( p \): Position
- \( v \): Velocity
- \( a \): Linear acceleration
- \( \omega \): Rotational velocity
- \( b_a \): Accel biases
- \( b_g \): Gyro biases
- \( g \): Gravity
- \( n_g \): Gyro meas/nt noise
- \( n_a \): Accel meas/nt noise
- \( n_{wg} \): Gyro bias process noise
- \( n_{wa} \): Accel bias process noise
**IMU Model**

- **IMU Measurement Model**
  - **Gyroscope:** \( \omega_m(t) = l' \omega(t) + b_g(t) + n_g(t) \)
  - **Accelerometer:** \( a_m(t) = C(l' q_G(t))(G a(t) - G g) + b_a(t) + n_a(t) \)

- **Continuous-time System Equations**
  \[
  \begin{align*}
  l' \dot{q}_G(t) &= \frac{1}{2} \Omega(\omega_m(t) - b_g(t) - n_g(t)) l' q_G(t) \\
  \dot{b}_g(t) &= n_{wg}(t) \\
  G \dot{v}_I(t) &= C^T(l' q_G(t))(a_m(t) - b_a(t) - n_a(t)) + G g \\
  \dot{b}_a(t) &= n_{wa}(t) \\
  G \dot{p}_I(t) &= G v_I(t)
  \end{align*}
  \]

- **IMU Integration** [1]
  \[
  x_{I_k} = f_d(x_{I_{k-1}}, u_{k-1:k}) + n_k
  \]

- **IMU Intrinsics** [2]
  - Accel/gyro scale factors & skewness
  - Accel-gyro relative orientation

---


Camera Model

• Camera Measurement Model

\[ z_i^j = h^{(C_i p_{f_j})} + n_i^j, \quad C_i p_{f_j} = C_i (C_i q_G)(C p_{f_j} - C p_{C_i}) \]
Camera Model

- Camera Measurement Model
  \[ z_i^j = h^{(C_i p_{f_j})} + n_i^j, \quad C_i p_{f_j} = C^{(C_i q_G)} (G p_{f_j} - G p_{C_i}) \]

- Camera Intrinsics
  - Principal point & focal length
  - Distortion parameters
  - Rolling-shutter time

Distorted image

Geometry change
Camera-IMU Model

- **Camera Measurement Model**
  \[ z_i^j = h(C_i p_{f_j}) + n_i^j, \quad C_i p_{f_j} = C(C_i q_G)(C p_{f_j} - C p_{C_i}) \]

- **Camera Intrinsics**
  - Principal point & focal length
  - Distortion parameters
  - Rolling-shutter time

- **Camera-IMU Extrinsic**s
  - Spatial: Rigid-body transformation \[^1\]
    \[ C_i p_{f_j} = C(G p_{f_j} - G p_{C_i}) = C(C_i q_G) C(G p_{f_j} - G p_{I_i}) + C I_i \]
  - Temporal: Time offset \[^2\]

  \[ C_z(x_i, G p_{f_j}, C q_I, t_d, t_r) = ||z_i^j - h(x_i, G p_{f_j}, C q_I, t_d, t_r)||^2_{\sigma_1} \]

---


Feature Extraction & Tracking

• Keypoint detection
  • Harris [1], DoG, FAST [2]
Feature Extraction & Tracking

- **Keypoint detection**
  - Harris \[1\], DoG, FAST \[2\]

- **Descriptor extraction**
  - SIFT\[3\], SURF\[4\], ORB\[5\], FREAK\[6\], BRISK\[7\], SDC\[8\]
Feature Extraction & Tracking

- Keypoint detection
  - Harris \cite{1}, DoG, FAST \cite{2}

- Descriptor extraction
  - SIFT\cite{3}, SURF\cite{4}, ORB\cite{5}, FREAK\cite{6}, BRISK\cite{7}, SDC\cite{8}

- Feature tracking (2D-to-2D)
  - KLT \cite{9}

\cite{9} B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” International Joint Conference on Artificial Intelligence’88
Feature Extraction & Tracking

• Keypoint detection
  • Harris [1], DoG, FAST [2]

• Descriptor extraction
  • SIFT [3], SURF [4], ORB [5], FREAK [6], BRISK [7], SDC [8]

• Feature tracking (2D-to-2D)
  • KLT [9]
  • Descriptor-to-descriptor matching
Feature Extraction & Tracking

- Keypoint detection
  - Harris [1], DoG, FAST [2]

- Descriptor extraction
  - SIFT [3], SURF [4], ORB [5], FREAK [6], BRISK [7], SDC [8]

- Feature tracking (2D-to-2D)
  - KLT [9]
  - Descriptor-to-descriptor matching
  - Outlier rejection (RANSAC)
    - w/out gyro: 5pt RANSAC [10]

Feature Extraction & Tracking

- Keypoint detection
  - Harris [1], DoG, FAST [2]
- Descriptor extraction
  - SIFT [3], SURF [4], ORB [5], FREAK [6], BRISK [7], SDC [8]
- Feature tracking (2D-to-2D)
  - KLT [9]
  - Descriptor-to-descriptor matching
  - Outlier rejection (RANSAC)
    - w/out gyro: 5pt RANSAC [10]

Loop-closure Detection

- Appearance-based image matching \[^1\]
  - Create image descriptor from feature descriptors
  - Compare image descriptors against each other

Loop-closure Detection

- Appearance-based image matching [1]
  - Create image descriptor from feature descriptors
  - Compare image descriptors against each other

Loop-closure Detection

- Appearance-based image matching \cite{1}
  - Create image descriptor from feature descriptors
  - Compare image descriptors against each other
- Outlier rejection / Geometric verification
  - 5pt\textsuperscript{2} (3pt+1\textsuperscript{3}) RANSAC to verify 2D-2D matches
  - P3P\textsuperscript{4} (P2+1\textsuperscript{5}) RANSAC for 2D-3D matches
  - Confirm loop-closure by matching consecutive images
    - Reduces false-positives
    - Delays map-based updates

\cite{2} D. Nister, “An efficient solution to the five-point relative pose problem,” TPAMI’04
\cite{3} O. Naroditsky, X. Zhou, S. Roumeliotis, and K. Daniilidis, “Two efficient solutions for visual odometry using directional correspondence,” TPAMI’12
\cite{4} T. Ke, S. Roumeliotis, “An Efficient Algebraic Solution to the Perspective-Three-Point Problem,” CVPR’17
\cite{5} Z. Kukelova, M. Bujnak, and T. Pajdla, “Closed-form solutions to minimal absolute pose problems with known vertical direction,” ACCV’11
Loop-closure Detection

- Appearance-based image matching [1]
  - Create image descriptor from feature descriptors
  - Compare image descriptors against each other
- Outlier rejection / Geometric verification
  - 5pt\textsubscript{2} (3pt+1\textsubscript{3}) RANSAC to verify 2D-2D matches
  - P3P\textsubscript{4} (P2+1\textsubscript{5}) RANSAC for 2D-3D matches
  - Confirm loop-closure by matching consecutive images
    - Reduces false-positives
    - Delays map-based updates

---

Sensor (IMU+Camera) Fusion

• Incremental BLS optimization\[1\]

\[ C_N = \sum_{k=1}^{N} \|x_{I_k} - f(x_{I_{k-1}}, u_{k-1:k})\|_Q^2 + \sum_{k=1}^{N} \sum_{j} \|z_{j}^i - h(x_{I_k}, G_p f_j)\|_{\sigma^2 I}^2 \]

• Issue: Memory/CPU req.s increase w/ time

Sensor (IMU+Camera) Fusion

- Incremental BLS optimization\(^1\)

\[
C_N = \sum_{k=1}^{N} \|x_{I_k} - f(x_{I_{k-1}}, u_{k-1:k})\|_Q^2 + \sum_{k=1}^{N} \sum_{j} \|z_{j}^k - h(x_{I_k}, G^j p^j_k)\|_I^2
\]

- Issue: Memory/CPU req.s increase w/ time
- Remedy: C-KLAM\(^2\) consistently marginalizes keyframes/features

Sensor (IMU+Camera) Fusion

- Incremental BLS optimization\[^1\]

\[
C_N = \sum_{k=1}^{N} \|x_{I_k} - f(x_{I_{k-1}}, u_{k-1:k})\|^2_Q + \sum_{k=1}^{N} \sum_{j} \|z^j_k - h(x_{I_k}, Gp_{f_j})\|^2_I
\]

- Issue: Memory/CPU req.s increase w/ time

- Alternative VINS approach
  - Split the problem into
    - Frontend (Localization): Fast, but drifts w/ time
      - e.g., Visual Inertial Odometry (VIO)
Sensor (IMU+Camera) Fusion

• Incremental BLS optimization\cite{1}

\[
C_N = \sum_{k=1}^{N} \|x_{I_k} - f(x_{I_{k-1}}, u_{k-1:k})\|^2_Q + \sum_{k=1}^{N} \sum_{j} \|z^j_k - h(x_{I_k}, G p_{f_j})\|^2_{\sigma^2_I}
\]

• Issue: Memory/CPU req.s increase w/ time

• Alternative VINS approach
  • Split the problem into
    • Frontend (Localization): Fast, but drifts w/ time
      • e.g., Visual Inertial Odometry (VIO)
    • Backend (Mapping): Slow, but more accurate
      • e.g., BLS, pose graph
Sensor (IMU+Camera) Fusion

• Incremental BLS optimization\textsuperscript{[1]}

\[ C_N = \sum_{k=1}^{N} \| x_{I_k} - f(x_{I_{k-1}}, u_{k-1:k}) \|_{Q_k}^2 + \sum_{k=1}^{N} \sum_{j} \| z^j_k - h(x_{I_k}, Gp_{f_j}) \|_{\sigma^2 I}^2 \]

• Issue: Memory/CPU req.s increase w/ time

• Alternative VINS approach
  • Split the problem into
    • Frontend (Localization): Fast, but drifts w/ time
      • e.g., Visual Inertial Odometry (VIO)
    • Backend (Mapping): Slow, but more accurate
      • e.g., BLS, pose graph
  • Relocalize w/ loop closures
    • Assumes keyframes of backend as perfectly known -> inconsistency
      • Estimated covariance < true covariance
Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

• **State Vector**

  \[ x_k = \begin{bmatrix} x_{I_k} & x_{I_{k-1}} & I_{k-2} \bar{q}_G & \cdots & I_{k-N} \bar{q}_G & C \bar{q}_I \end{bmatrix}^T \]

  where \( x_I = \begin{bmatrix} I q_G^T & b_g^T & G v_I^T & b_a^T & G p_I^T \end{bmatrix}^T \)

  \[ I \bar{q}_G = \begin{bmatrix} I q_G^T & G p_I^T \end{bmatrix}^T \]

Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

• Step 1: Propagation

\[ C_{k|k-1} = \underbrace{\|x_{k-1}\|_P^{2}_{k-1}}_{\text{Prior}} + \underbrace{\|x_{I_k} - f(x_{I_{k-1}}, u_{k-1:k})\|_Q^{2}}_{\text{IMU}} \]
Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

• Step 1: Propagation

\[
C_{k|k-1} = \underbrace{\|\tilde{x}_{k-1}\|^2_{P_{k-1|k-1}}}_{Prior} + \underbrace{\|x_{I_k} - f(x_{I_{k-1}}, u_{k-1:k})\|^2_{Q_k}}_{IMU} \\
C_{k|k-1} \simeq \underbrace{\|\tilde{x}_k\|^2_{P_{k|k-1}}}_{New\ Prior}
\]
Frontend: Multi-state Constraint Kalman Filter (MSCKF) \[1\]

- **Step 1: Propagation**
  \[
  C_{k|k-1} = \left(\|\bar{x}_{k-1}\| P_{k-1|k-1}^2 + \|x_{I_k} - f(x_{I_k-1}, u_{k-1:k})\|^2 \right)_{Q_k}^{\text{Prior}} + \|x_{I_k} - f(x_{I_k-1}, u_{k-1:k})\|^2 \quad \text{IMU}
  \]
  \[\]
  \[
  C_{k|k-1} \simeq \|\bar{x}_k\|^2 P_{k|k-1} \quad \text{New Prior}
  \]

- **Step 2: Marginalize all features \([O(N)]\)**
  \[
  C_{k|k} = \|\bar{x}_k\|^2 P_{k|k-1} + \sum_j \|z_{k-N:k}^j - f(x_{k-N:k}, p_{f_j})\|^2
  \]
  \[
  \simeq \|\bar{x}_k\|^2 P_{k|k-1} + \sum_j \|z_{k-N:k}^j - (H_x^j \tilde{x}_{k-N:k} + H_{f_j}^j \tilde{p}_{f_j})\|^2
  \]
  \[
  = \|\bar{x}_k\|^2 P_{k|k-1} + \sum_j \left(\|U_j^T \tilde{z}_{k-N:k}^j - U_j^T H_x^j \tilde{x}_{k-N:k}\|^2 + \|V_j^T \tilde{z}_{k-N:k}^j - (V_j^T H_x^j \tilde{x}_{k-N:k} + V_j^T H_{f_j}^j \tilde{p}_{f_j})\|^2\right)
  \]
  where \([V_j \ U_j]^T [V_j \ U_j] = I, \ U_j^T H_{f_j}^j = 0\)
Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

- **Step 1: Propagation**

  \[
  C_{k|k-1} = \underbrace{\|\tilde{x}_{k-1}\|_{\tilde{P}_{k-1|k-1}}^2} + \underbrace{\|x_{I_k} - f(x_{I_k}, u_{k-1:k})\|_{Q_k}} \tag{Prior IMU}
  \]

  \[
  C_{k|k-1} \approx \|\tilde{x}_k\|_{\tilde{P}_{k|k-1}}^2 \tag{New Prior}
  \]

- **Step 2: Marginalize all features \([O(N)]\)**

  \[
  C_{k|k} = \|\tilde{x}_k\|_{\tilde{P}_{k|k-1}}^2 + \sum_j \|z_{k-N:k}^j - f(x_{k-N:k}, p_{f_j})\|^2
  \]

  \[
  \approx \|\tilde{x}_k\|_{\tilde{P}_{k|k-1}}^2 + \sum_j \|\tilde{z}_{k-N:k}^j - (H_x^j \tilde{x}_{k-N:k} + H_f^j \tilde{p}_{f_j})\|^2
  \]

  \[
  = \|\tilde{x}_k\|_{\tilde{P}_{k|k-1}}^2 + \sum_j \left(\|U_j^T \tilde{z}_{k-N:k}^j - U_j^T H_x^j \tilde{x}_{k-N:k}\|^2 + \|V_j^T \tilde{z}_{k-N:k}^j - (V_j^T H_p^j \tilde{x}_{k-N:k} + V_j^T H_f^j \tilde{p}_{f_j})\|^2\right)
  \]

  \[
  C'_{k|k} = \|\tilde{x}_k\|_{\tilde{P}_{k|k-1}}^2 + \sum_j \|U_j^T \tilde{z}_{k-N:k}^j - U_j^T H_x^j \tilde{x}_{k-N:k}\|^2
  \]

  \[
  = \|\tilde{x}_k\|_{\tilde{P}_{k|k-1}}^2 + \|\tilde{z}_{k-N:k} - H_x \tilde{x}_{k-N:k}\|^2
  \]

  \[
  \text{where } [V_j U_j]^T [V_j U_j] = I, \quad U_j^T H_f^j = 0
  \]
Frontend: Multi-state Constraint Kalman Filter (MSCKF) \[1\]

• Step 3: Update \(O(M^3)\)

\[
C'_{k|k} = \frac{1}{2} \left( \|\tilde{x}_k\|^2_{P_{k|k-1}} + \|\tilde{z}_{k-N:k} - H_x \tilde{x}_{k-N:k}\|^2 \right)
= \frac{1}{2} \|\tilde{x}_{k-N:k}\|^2_{P_{k-N:k}^\oplus}
\]
Frontend: Multi-state Constraint Kalman Filter (MSCKF) \[1\]

- Step 3: Update \([O(M^3)]\)

\[ C'_{k|k} = \|\mathbf{x}_k\|_{P_{k|k-1}}^2 + \|\mathbf{z}_{k-N:k} - \mathbf{H}_x \mathbf{x}_{k-N:k}\|_2^2 \]

\[ = \|\mathbf{x}_{k-N:k}\|_{P_{k-N:k}}^{2} \]

- Step 4: Marginalize the oldest pose \(I_{k-N}\)

\[ C'_{k|k} = \|\mathbf{x}_{k-N+1:k}\|_{P_{k-N+1:k}}^{2} + \|L_1 I_{k-N} \mathbf{q}_G + L_2 \mathbf{x}_{k-N+1:k}\|_2^2 \]
Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

• Step 3: Update $[O(M^3)]$

\[
C'_{k|k} = \frac{1}{P_{k|k-1}} + \frac{1}{H_x \tilde{x}_{k-N:k}^2}
\]

\[
= \frac{1}{P_{k-N:k}^\oplus}
\]

• Step 4: Marginalize the oldest pose $I_{k-N} \tilde{q}_G$

\[
C''_{k|k} = \frac{1}{P_{k-N:1:k}^\oplus}
\]

\[
C_{k|k} = \frac{1}{P_{k-N:1:k}^\oplus}
\]

\[
\frac{1}{P_{k-N:1:k}^\oplus} + \frac{1}{L_1 I_{k-N} \tilde{q}_G + L_2 \tilde{x}_{k-N+1:k}^2}
\]

\[
\text{Posterior}
\]
MSCKF Feature Classification & Processing

- Mature feature: Track starts at the oldest pose (to be marginalized)
  - Track spans part of the window -> Marginalize w/ **MSCKF**
MSCKF Feature Classification & Processing

- Mature feature: Track starts at the oldest pose (to be marginalized)
  - Track spans part of the window -> Marginalize w/ **MSCKF**
  - Track spans the whole window -> Add to the state vector as **SLAM** feature
MSCKF Feature Classification & Processing

• Mature feature: Track starts at the oldest pose (to be marginalized)
  • Track spans part of the window -> Marginalize w/ **MSCKF**
  • Track spans the whole window -> Add to the state vector as **SLAM** feature

• Immature feature: Track is still ongoing
  • Use as **state-only** feature (update states, but not covariance)
Filtering vs. Optimization-based Methods

- **MSCKF (EKF) ↔ MAP estimator w/ one Gauss-Newton iteration** \([1]\)
  - Iteratively processes camera meas/nts (Iterated EKF \([2]\)), IMU meas/nts (IKS \([3]\))
- **MSCKF (EKF) ↔ SWF (EIF)** \([4]\)
- **Square-root variants: SR-EKF, SR-EIF** \([5]\)
  - Use Cholesky factor of covariance/Hessian
  - Better numerical properties
  - Single-precision arithmetic (4x speed up for ARM Neon coprocessor)

---

<table>
<thead>
<tr>
<th>Prior</th>
<th>Covariance ( \mathbf{P} )</th>
<th>Information ( \mathbf{A} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propagation</td>
<td>( \mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} &amp; \mathbf{P}\Phi \ \Phi\mathbf{P} &amp; \Phi\mathbf{P}\Phi^T + \mathbf{W} \end{bmatrix} )</td>
<td>( \mathbf{A} \leftarrow \begin{bmatrix} \mathbf{A} + \Phi^T \mathbf{W}^{-1} \Phi &amp; -\Phi^T \mathbf{W}^{-1} \ -\mathbf{W}^{-1} \Phi &amp; \mathbf{W}^{-1} \end{bmatrix} )</td>
</tr>
<tr>
<td>( \mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{\Sigma} )</td>
<td>( \mathbf{P}^\oplus = \mathbf{P} - \mathbf{P}\mathbf{H}^T \mathbf{S}^{-1} \mathbf{HP} )</td>
<td>( \mathbf{b}^\oplus = \mathbf{H}^T \mathbf{S}^{-1} \mathbf{r} )</td>
</tr>
<tr>
<td>( \mathbf{P} \leftarrow \mathbf{P}^\oplus )</td>
<td>( \Delta \mathbf{x} = \mathbf{PH}^T \mathbf{S}^{-1} \mathbf{r} )</td>
<td>( \Delta \mathbf{x} = \mathbf{A}^{-1} \mathbf{b} )</td>
</tr>
<tr>
<td>( \mathbf{x}^\oplus = \mathbf{x} + \Delta \mathbf{x} )</td>
<td>( \mathbf{x} \leftarrow \begin{bmatrix} \mathbf{x} \ \mathbf{x}_\nu \end{bmatrix} )</td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Marginalization</th>
<th>( \Pi \mathbf{P}^T = \begin{bmatrix} \mathbf{P}<em>{\mu\mu} &amp; \mathbf{P}</em>{\mu\rho} \ \mathbf{P}<em>{\rho\mu} &amp; \mathbf{P}</em>{\rho\rho} \end{bmatrix} )</th>
<th>( \Pi \mathbf{A} \mathbf{P}^T = \begin{bmatrix} \mathbf{A}<em>{\mu\mu} &amp; \mathbf{A}</em>{\mu\rho} \ \mathbf{A}<em>{\rho\mu} &amp; \mathbf{A}</em>{\rho\rho} \end{bmatrix} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{P} \leftarrow \mathbf{P}_{\rho\rho} )</td>
<td>( \mathbf{A}'<em>{\rho\rho} = \mathbf{A}</em>{\rho\rho} - \mathbf{A}<em>{\rho\mu} \mathbf{A}</em>{\mu\mu}^{-1} \mathbf{A}_{\mu\rho} )</td>
<td>( \mathbf{A} \leftarrow \mathbf{A}'_{\rho\rho} )</td>
</tr>
</tbody>
</table>

---

Inconsistency of VIO

Due to *mismatch* of observability properties btwn nonlinear system and linearized estimator [1,2,3,4]

**Inconsistency of VIO**

Due to *mismatch* of observability properties between nonlinear system and linearized estimator [1,2,3,4]

Actual nonlinear system:
\[
\begin{align*}
\dot{x} &= f_0(x) + \sum_{i=1}^l f_i(u) \\
z &= h(x)
\end{align*}
\]

Ideal linearized system:
\[
\begin{align*}
x_{k+1|k} &= \Phi_{k|k} x_k|k + G_{k|k} u_k \\
z_k &= H_{k|k} \hat{x}_k|k
\end{align*}
\]

Due to mismatch of observability properties between nonlinear system and linearized estimator.

- Inf. dim.
- Full col. rank
- Observable

Global transformation:

- Rot. around gravity

Null(\(\mathcal{O}\))\(_{VIO}\) =

\[
\begin{bmatrix}
0_3 & \frac{I_c}{G} \cdot C^{G} g \\
0_3 & 0_3 \\
0_3 & -[v \times ]^G g \\
0_3 & 0_3 \\
I_3 & -[p \times ]^G g \\
I_3 & [p_f \times ]^G g
\end{bmatrix}
\]

\[
\begin{bmatrix}
\theta \\
b_g \\
v \\
b_n \\
p \\
p_f
\end{bmatrix} = \begin{bmatrix}
N_t \\
N_r
\end{bmatrix}
\]


Inconsistency of VIO

Due to mismatch of observability properties btwn nonlinear system and linearized estimator [1,2,3,4]

Actual nonlinear system
\[
\begin{align*}
\dot{x} &= f_0(x) + \sum_{i=1}^{l} f_i(u) \\
z &= h(x)
\end{align*}
\]

Ideal linearized system
\[
\begin{align*}
x_{k+1|k} &= \Phi_k|x_{k|k} + G_k|k u_k \\
z_k &= H_k|x_{k|k}
\end{align*}
\]

Actual linearized estimator
\[
\begin{align*}
\hat{x}_{k+1|k} &= \hat{\Phi}_k|k \hat{x}_{k|k} + \hat{G}_k|k u_k \\
\hat{z}_k &= \hat{H}_k|k \hat{x}_{k|k}
\end{align*}
\]

\[\mathcal{O} = \begin{bmatrix}
\nabla^0 h \\
\nabla^1 f_1 h \\
\nabla^1 f_2 h \\
\vdots
\end{bmatrix} \Rightarrow \begin{bmatrix}
\frac{\partial g^0 h}{\partial \beta} \\
\frac{\partial g^1 f_1 h}{\partial \beta} \\
\frac{\partial g^2 f_2 h}{\partial \beta} \\
\vdots
\end{bmatrix} = \begin{bmatrix}
\beta
\end{bmatrix}
\]

Global trans.
- Inf. dim.
- Full col. rank
- Observable

null(\mathcal{O}) = \text{Null}(B)

Null(\mathcal{O})_{VIO} = \begin{bmatrix}
0_3 & l_G C^G g \\
0_3 & 0_3 \\
0_3 & -[v \times]^G g \\
0_3 & 0_3 \\
I_3 & -[p \times]^G g \\
I_3 & [p_f \times]^G g
\end{bmatrix} \theta \begin{bmatrix}
b_g \\
v \\
b_n \\
p \\
p_f
\end{bmatrix} = [N_t \mid N_r]

Rot. around gravity

\[\text{Null}(\mathcal{M}) = \text{Null}(\mathcal{O})_{VIO} \subsetneq \text{Null}(\hat{\mathcal{M}}) = N_t\]

Inconsistency of VIO

Due to \textit{mismatch} of observability properties btwn nonlinear system and linearized estimator [1,2,3,4]

Actual nonlinear system

\[
\begin{align*}
\dot{x} &= f_0(x) + \sum_{i=1}^{l} f_i(u) \\
z &= h(x)
\end{align*}
\]

\[O = \begin{bmatrix} \nabla \frac{\partial g^0_h}{\partial \beta} \\
\nabla \frac{\partial g^1_h}{\partial \beta} \\
\cdots \\
\nabla \frac{\partial g^3_{f_t f_t}}{\partial \beta} \\
\end{bmatrix} = \begin{bmatrix} \begin{array}{c}
\frac{\partial g^0_h}{\partial \beta} \\
\frac{\partial g^1_h}{\partial \beta} \\
\cdots \\
\frac{\partial g^3_{f_t f_t}}{\partial \beta} \\
\end{array}
\end{bmatrix}
\]

\[
\frac{\partial \beta}{\partial x} = \Theta = \begin{bmatrix} 0 & I \end{bmatrix}
\]

- Inf. dim.
- Full col. rank
- Observable

Global trans.

Rot. around gravity

\[\text{Null}(O)_{VIO} = \begin{bmatrix} 0_3 & I \end{bmatrix} \begin{bmatrix} G \cdot g \\
0_3 \\
v \times [G \cdot g] \\
0_3 \\
p \times [g_f] \\
0_3 \\
p \times [f] \\
\end{bmatrix} = \begin{bmatrix} N_t \\
N_r \\
\end{bmatrix} \]

Ideal linearized system

\[
\begin{align*}
x_{k+1|k} &= \Phi_{k|k} x_{k|k} + G_{k|k} u_k \\
z_k &= H_{k|k} x_{k|k}
\end{align*}
\]

\[
M = \begin{bmatrix} H_k & \Phi_{k+1,k|k} \\
H_{k+1,k|k} & \Phi_{k+2,k|k} \\
\vdots & \vdots \\
H_{k+m,k|k} & \Phi_{k+m+1,k+1|k}
\end{bmatrix}
\]

Actual linearized estimator

\[
\begin{align*}
\hat{x}_{k+1|k} &= \hat{\Phi}_{k|k} \hat{x}_{k|k} + \hat{G}_{k|k} u_k \\
\hat{z}_k &= \hat{H}_{k|k} \hat{x}_{k|k}
\end{align*}
\]

\[
\hat{M} = \begin{bmatrix} \hat{H}_k & \hat{\Phi}_{k+1,k|k} \\
\hat{H}_{k+1,k|k} & \hat{\Phi}_{k+2,k|k} \\
\vdots & \vdots \\
\hat{H}_{k+m,k|k} & \hat{\Phi}_{k+m+1,k+1|k}
\end{bmatrix}
\]

\[\text{Null}(M) = \text{Null}(O)_{VIO} \supset \text{Null}(\hat{M}) = N_t\]


Mapping Backend

• Offline: BA [1,2], CM [3]

Mapping Backend

• Offline: BA\textsuperscript{[1,2]}, CM\textsuperscript{[3]}
• Online
  • BLS Approximation: PTAM\textsuperscript{[4]}, iSAM2\textsuperscript{[5]}, C-KLAM\textsuperscript{[6]}
    • Employ approximations e.g., perfect keyframe/feature assumption, delay relinearization, duplicate meas/nts

\[\text{ICRA'16}\]

Mapping Backend

- Offline: BA\textsuperscript{[1,2]}, CM\textsuperscript{[3]}
- Online
  - BLS Approximation: PTAM\textsuperscript{[4]}, iSAM2\textsuperscript{[5]}, C-KLAM\textsuperscript{[6]}
    - Employ approximations e.g., perfect keyframe/feature assumption, delay relinearization, duplicate meas/nts
  - Sub-mapping: Tectonic SAM\textsuperscript{[7]}, Gravity aligned sub-maps\textsuperscript{[8]}
    - Divide map into submaps and merge

\textsuperscript{[7]} K. Ni, D. Steedly, and F. Dellaert, “Tectonic SAM: Exact, out-of-core, submap-based SLAM,” ICRA’07
\textsuperscript{[8]} K. Sartipi and S. Roumeliotis, “Efficient alignment of visual-inertial maps,” ISER’18
Mapping Backend

- Offline: BA\[^{[1,2]}\], CM\[^{[3]}\]
- Online
  - BLS Approximation: PTAM\[^{[4]}\], iSAM2\[^{[5]}\], C-KLAM\[^{[6]}\]
    - Employ approximations e.g., perfect keyframe/feature assumption, delay relinearization, duplicate meas/nts
  - Sub-mapping: Tectonic SAM\[^{[7]}\], Gravity aligned sub-maps\[^{[8]}\]
    - Divide map into submaps and merge
  - Pose-graph: Gutmann and Konolige\[^{[9]}\], GraphSLAM\[^{[10]}\], VINS-Mono\[^{[11]}\]
    - Use features to determine relative poses and optimize only for poses

---

Map-based Updates

• Mapped keyframes/key features provided by the Backend to the Frontend
Map-based Updates

- Mapped keyframes/key features provided by the Backend to the Frontend
- Map assumed perfectly known [1,2]

Map-based Updates

- Mapped keyframes/key features provided by the Backend to the Frontend
- Map assumed perfectly known \[^{1,2}\]  
  - Advantage: Constant processing cost
  - Disadvantage: Inconsistent

\[ C_{map} = \| z_k^j - h(x_k, p_{f_j}) \|^2_{\sigma^2 I} \approx \| z_k^j - H_k^j \tilde{x}_k - H_k^j c_{f_j} \|^2_{\sigma^2 I} \]

Map-based Updates

- Mapped keyframes/key features provided by the Backend to the Frontend

- Map assumed perfectly known \[^1,2\]
  - Advantage: Constant processing cost
  - Disadvantage: Inconsistent
    - Remedy: Inflate meas/mnt noise

\[
C_{map} = \left\| z_k^j - h(x_k, p_{f, j}) \right\|^2_{\sigma^2 I} \simeq \left\| \tilde{z}_k^j - H_k^j \tilde{x}_k - H_{f, j}^j b_{f, j} \right\|^2_{4\sigma^2 I}
\]


Map-based Updates

- Mapped keyframes/key features provided by the Backend to the Frontend
- Map assumed perfectly known\[^{1,2}\]  
  - Advantage: Constant processing cost  
  - Disadvantage: Inconsistent  
    - Remedy: Inflate meas/mnt noise

\[
C_{map} = \left\| z_k^j - h(x_k, p_{f_j}) \right\|^2_{\sigma_j^2 I} \simeq \left\| z_k^j - H_k^j \tilde{x}_k - h_j^j f_{j} \right\|^2_{4\sigma_j^2 I}
\]

- Consistent alternatives:  
  - Schmidt Kalman Filter \[^3\]  
  - RISE-SLAM \[^4\]

---

Cooperative VIO/SLAM

• Data from multiple devices are fused to create an area representation

• Centralized [1,2]
  • Computation is offloaded from device
  • Require powerful server for processing

Cooperative VIO/SLAM

• Data from multiple devices are fused to create an area representation

• Centralized \[1,2\]
  • Computation is offloaded from device
  • Require powerful server for processing

• Distributed \[3,4\]
  • All devices cooperate to compute a single area representation


Cooperative VIO/SLAM

- Data from multiple devices are fused to create an area representation
- Centralized [1,2]
  - Computation is offloaded from device
  - Require powerful server for processing
- Distributed [3,4]
  - All devices cooperate to compute a single area representation
- Multi-centralized [5,6,7]
  - Each device computes a map of the area

Interesting Research Directions

- **Observability Analysis**
  - Additional unobservable directions [1]
    - scale – under const. linear accel.
    - roll, pitch – under const. orientation

Interesting Research Directions

• **Observability Analysis**
  • Additional unobservable directions [1]
    • scale – under const. linear accel.
    • roll, pitch – under const. orientation
  • Types of Features: edges, lines, planes [2,3,4]
  • IMU/Camera intrinsics, extrinsics, RS, TS

• **Event-based camera** [5,6]
  • Detect changes in intensity, low latency

• **Incorporate system’s dynamics**
  • Human motion model [7]

---

• **Information selection**
  • **Geometry based (improve accuracy)**
    • Greedy and consider user’s intention [1]
    • Heuristics: Long tracks, uniformly distributed, wide baseline, close-by [2]
    • Multi-camera resource allocation [3]

• **Information selection**
  
  • *Geometry based (improve accuracy)*
    - Greedy and consider user’s intention [1]
    - Heuristics: Long tracks, uniformly distributed, wide baseline, close-by [2]
    - Multi-camera resource allocation [3]
  
  • *Semantic ([4,5] based (improve robustness)* exclude ephemeral parts of the scene

  **moving objects** concern filtering

  **movable objects** concern mapping

• **Robust scene recognition using ML features** [6]

  **Season/light Invariance**
  
  **Viewpoint Invariance**

---


Thank You!