Visual Inertial Navigation
Short Tutorial

Stergios Roumeliotis

University of Minnesota

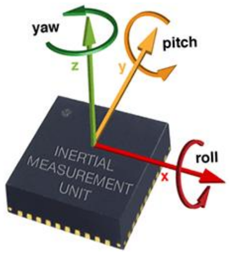
Outline

- 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

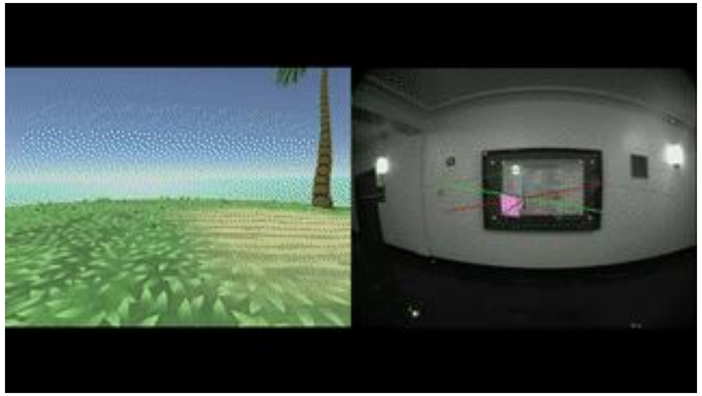
[1] G.P. Huang, "Visual-inertial navigation: A concise review," ICRA'19

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

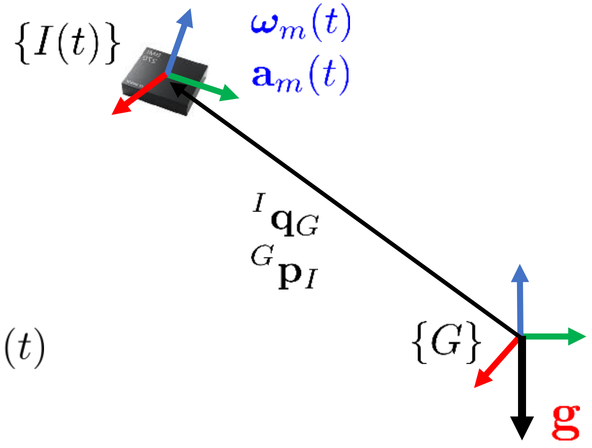


- Applications
  - Autonomous navigation, augmented/virtual reality



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

IMU Model



- IMU Measurement Model
  - Gyroscope: $\boldsymbol{\omega}_m(t) = {}^I\boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t)$
  - Accelerometer: $\mathbf{a}_m(t) = \mathbf{C}({}^I\mathbf{q}_G(t))({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t)$
- Continuous-time System Equations

$$\left.\begin{aligned}
{}^I\dot{\mathbf{q}}_G(t) &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_m(t) - \mathbf{b}_g(t) - \mathbf{n}_g(t))\,{}^I\mathbf{q}_G(t) \\
\dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}(t) \\
{}^G\dot{\mathbf{v}}_I(t) &= \mathbf{C}^T({}^I\mathbf{q}_G(t))(\mathbf{a}_m(t) - \mathbf{b}_a(t) - \mathbf{n}_a(t)) + {}^G\mathbf{g} \\
\dot{\mathbf{b}}_a(t) &= \mathbf{n}_{wa}(t) \\
{}^G\dot{\mathbf{p}}_I(t) &= {}^G\mathbf{v}_I(t)
\end{aligned}\right\}$$

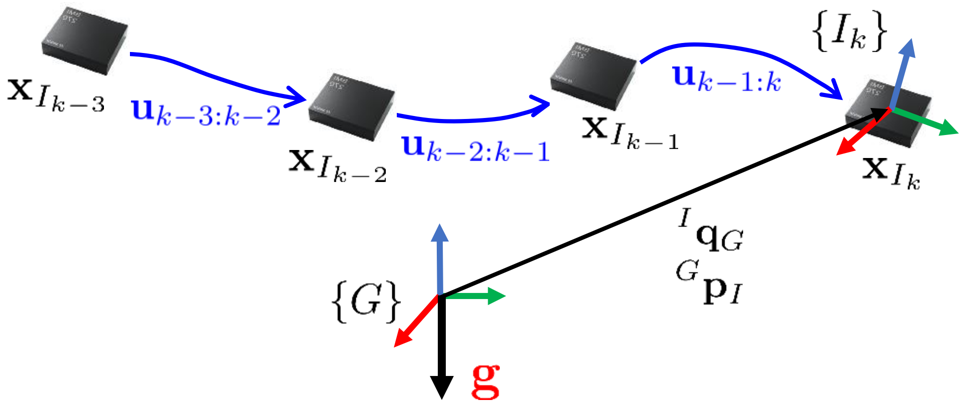
$$\dot{\mathbf{x}}_I(t) = \mathbf{f}_c(\mathbf{x}_I(t), \mathbf{u}(t) - \mathbf{n}(t))$$

$$\mathbf{x}_I = \begin{bmatrix} {}^I\mathbf{q}_G^T & \mathbf{b}_g^T & {}^G\mathbf{v}_I^T & \mathbf{b}_a^T & {}^G\mathbf{p}_I^T \end{bmatrix}^T$$

$$\mathbf{u}(t) = \begin{bmatrix} \boldsymbol{\omega}_m(t)^T & \mathbf{a}_m(t)^T \end{bmatrix}^T$$

- $\mathbf{q}$: Quaternion of orientation
- $\mathbf{C}$: Rotation matrix
- $\mathbf{p}$: Position
- $\mathbf{v}$: Velocity
- $\mathbf{a}$: Linear acceleration
- $\boldsymbol{\omega}$: Rotational velocity
- $\mathbf{b}_a$: Accel biases
- $\mathbf{b}_g$: Gyro biases
- $\mathbf{g}$: Gravity
- $\mathbf{n}_g$: Gyro meas/nt noise
- $\mathbf{n}_a$: Accel meas/nt noise
- $\mathbf{n}_{wg}$: Gyro bias process noise
- $\mathbf{n}_{wa}$: Accel bias process noise

IMU Model



- IMU Measurement Model
  - Gyroscope: $\boldsymbol{\omega}_m(t) = {}^I\boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t)$
  - Accelerometer: $\mathbf{a}_m(t) = \mathbf{C}({}^I\mathbf{q}_G(t))({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t)$
- Continuous-time System Equations

$$
\left.\begin{aligned}
{}^I\dot{\mathbf{q}}_G(t) &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_m(t) - \mathbf{b}_g(t) - \mathbf{n}_g(t))\,{}^I\mathbf{q}_G(t) \\
\dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}(t) \\
{}^G\dot{\mathbf{v}}_I(t) &= \mathbf{C}^T({}^I\mathbf{q}_G(t))(\mathbf{a}_m(t) - \mathbf{b}_a(t) - \mathbf{n}_a(t)) + {}^G\mathbf{g} \\
\dot{\mathbf{b}}_a(t) &= \mathbf{n}_{wa}(t) \\
{}^G\dot{\mathbf{p}}_I(t) &= {}^G\mathbf{v}_I(t)
\end{aligned}\right\}
\begin{aligned}
\dot{\mathbf{x}}_I(t) &= \mathbf{f}_c(\mathbf{x}_I(t), \mathbf{u}(t) - \mathbf{n}(t)) \\
\mathbf{x}_I &= \begin{bmatrix} {}^I\mathbf{q}_G^T & \mathbf{b}_g^T & {}^G\mathbf{v}_I^T & \mathbf{b}_a^T & {}^G\mathbf{p}_I^T \end{bmatrix}^T \\
\mathbf{u}(t) &= \begin{bmatrix} \boldsymbol{\omega}_m(t)^T & \mathbf{a}_m(t)^T \end{bmatrix}^T
\end{aligned}
$$

- IMU Integration [1]

$$\mathbf{x}_{I_k} = \mathbf{f}_d(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k}) + \mathbf{n}_k \Rightarrow \boxed{\mathcal{C}_u(\mathbf{x}_{I_{k-1}}, \mathbf{x}_{I_k}) = \|\mathbf{x}_{I_k} - \mathbf{f}_d(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k}}$$

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

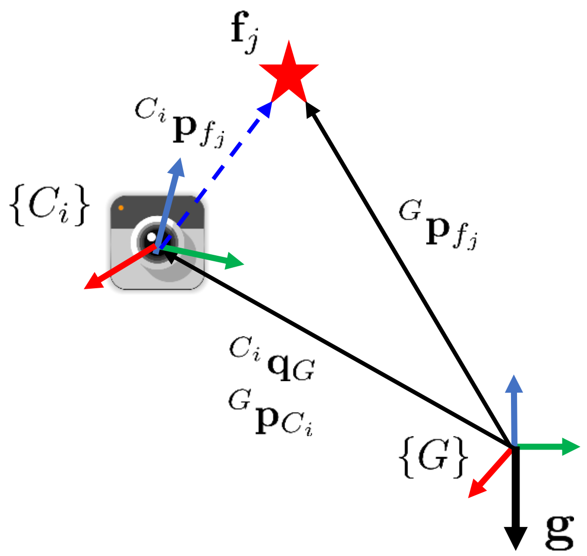
$\mathbf{q}$: Quaternion of orientation
$\mathbf{C}$: Rotation matrix
$\mathbf{p}$: Position
$\mathbf{v}$: Velocity
$\mathbf{a}$: Linear acceleration
$\boldsymbol{\omega}$: Rotational velocity
$\mathbf{b}_a$: Accel biases
$\mathbf{b}_g$: Gyro biases
$\mathbf{g}$: Gravity
$\mathbf{n}_g$: Gyro meas/nt noise
$\mathbf{n}_a$: Accel meas/nt noise
$\mathbf{n}_{wg}$: Gyro bias process noise
$\mathbf{n}_{wa}$: Accel bias process noise

[1] A. I. Mourikis and S. I. Roumeliotis, "**A multi-state constraint Kalman filter for vision-aided inertial navigation**," ICRA'07
[2] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "**High-fidelity sensor modeling and calibration in vision-aided inertial navigation**," ICRA'14

Camera Model

- Camera Measurement Model

$$\mathbf{z}_i^j = \mathbf{h}({}^{C_i}\mathbf{p}_{f_j}) + \mathbf{n}_i^j \ , \qquad {}^{C_i}\mathbf{p}_{f_j} = \mathbf{C}({}^{C_i}\mathbf{q}_G)({}^{G}\mathbf{p}_{f_j} - {}^{G}\mathbf{p}_{C_i})$$

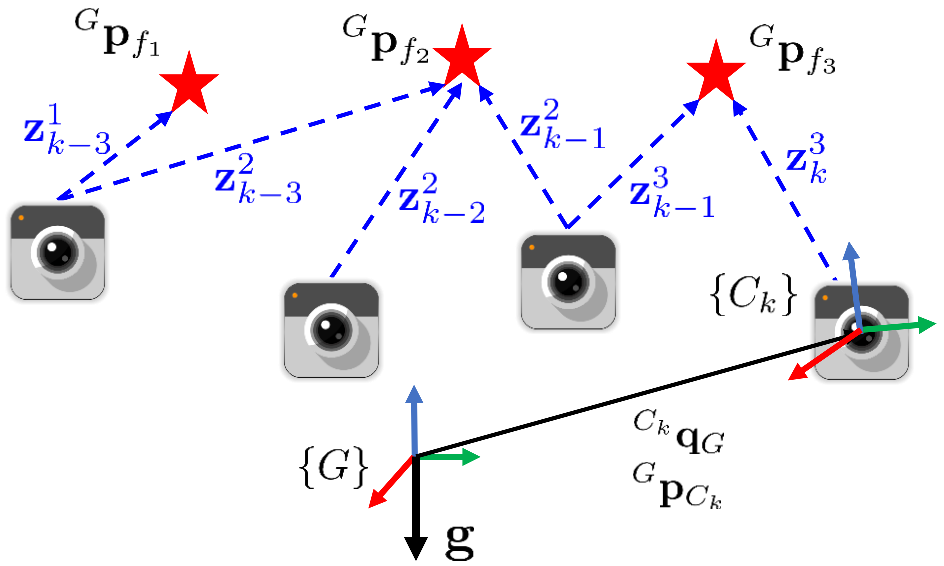


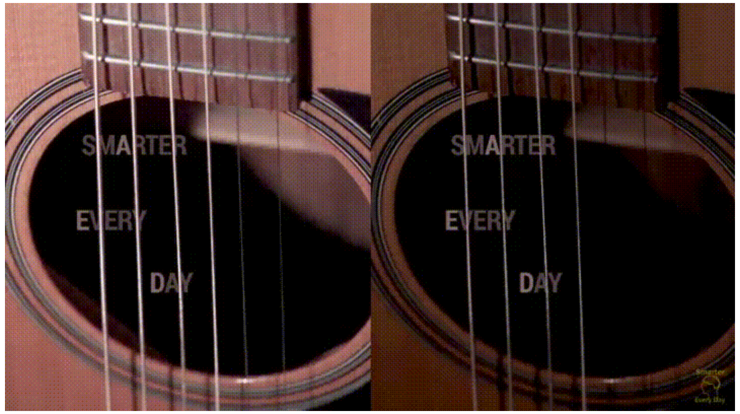
Camera Model

- Camera Measurement Model

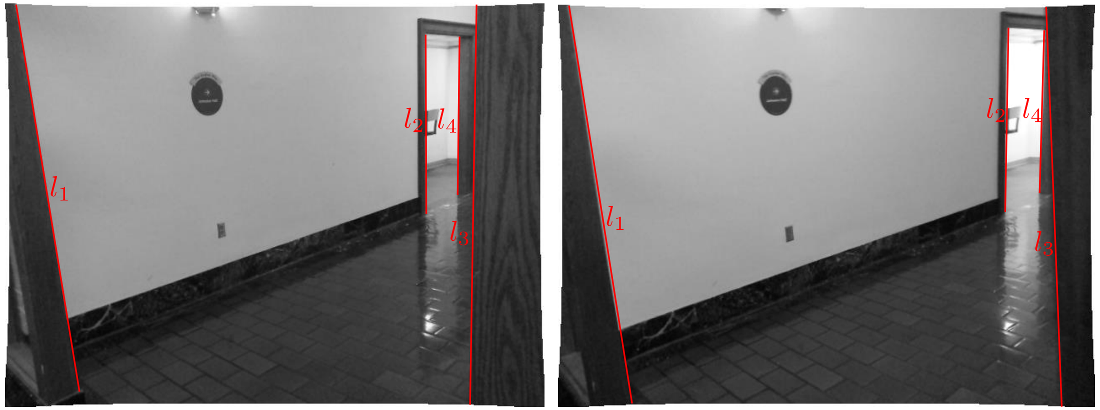
$$\mathbf{z}_i^j = \mathbf{h}({}^{C_i}\mathbf{p}_{f_j}) + \mathbf{n}_i^j \ , \qquad {}^{C_i}\mathbf{p}_{f_j} = \mathbf{C}({}^{C_i}\mathbf{q}_G)({}^{G}\mathbf{p}_{f_j} - {}^{G}\mathbf{p}_{C_i})$$

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





Distorted image



Geometry change

Camera-IMU Model

- Camera Measurement Model

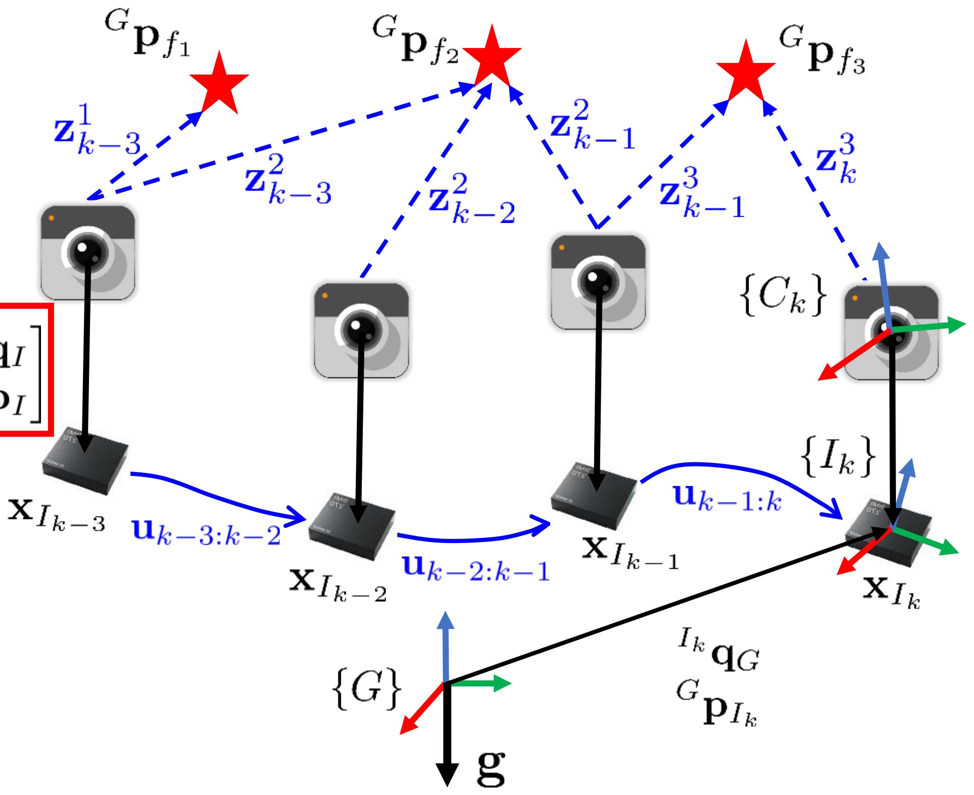
$$\mathbf{z}_i^j = \mathbf{h}({}^{C_i}\mathbf{p}_{f_j}) + \mathbf{n}_i^j \,, \qquad {}^{C_i}\mathbf{p}_{f_j} = \mathbf{C}({}^{C_i}\mathbf{q}_G)({}^G\mathbf{p}_{f_j} - {}^G\mathbf{p}_{C_i})$$

- Camera Intrinsics
  - Principal point & focal length
  - Distortion parameters
  - Rolling-shutter time
- Camera-IMU Extrinsics
  - Spatial: Rigid-body transformation [1]

$${}^{C_i}\mathbf{p}_{f_j} = {}^{C_i}_{G}\mathbf{C}({}^G\mathbf{p}_{f_j} - {}^G\mathbf{p}_{C_i}) = \boxed{\mathbf{C}({}^C\mathbf{q}_I)}\,{}^{I_i}_{G}\mathbf{C}({}^G\mathbf{p}_{f_j} - {}^G\mathbf{p}_{I_i}) + \boxed{{}^C\mathbf{p}_I}$$

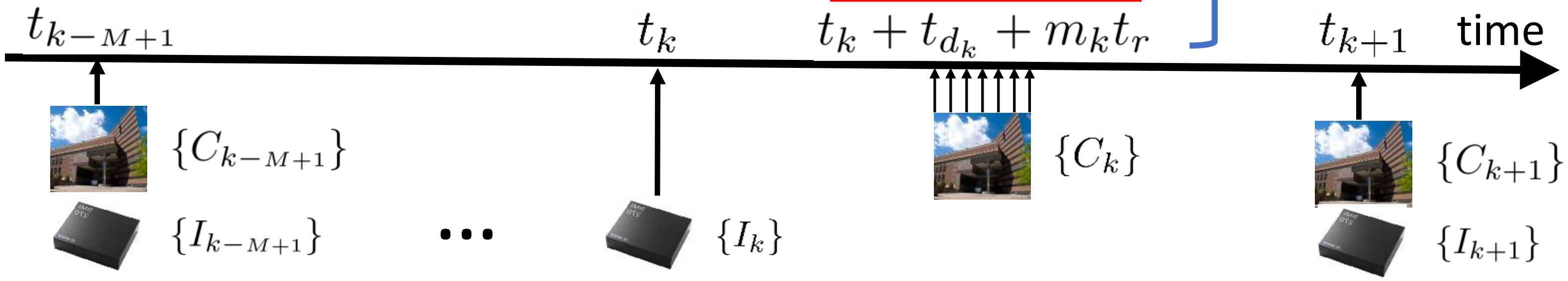
  - Temporal: Time offset [2]

$$\boxed{{}^C\bar{\mathbf{q}}_I = \begin{bmatrix} {}^C\mathbf{q}_I \\ {}^C\mathbf{p}_I \end{bmatrix}}$$



$$\mathcal{C}_z(\mathbf{x}_{I_i}, {}^G\mathbf{p}_{f_j}, {}^C\bar{\mathbf{q}}_I, t_{d_i}, t_r) = \|\mathbf{z}_i^j - \mathbf{h}(\mathbf{x}_{I_i}, {}^G\mathbf{p}_{f_j}, {}^C\bar{\mathbf{q}}_I, t_{d_i}, t_r)\|^2_{\sigma^2\mathbf{I}}$$

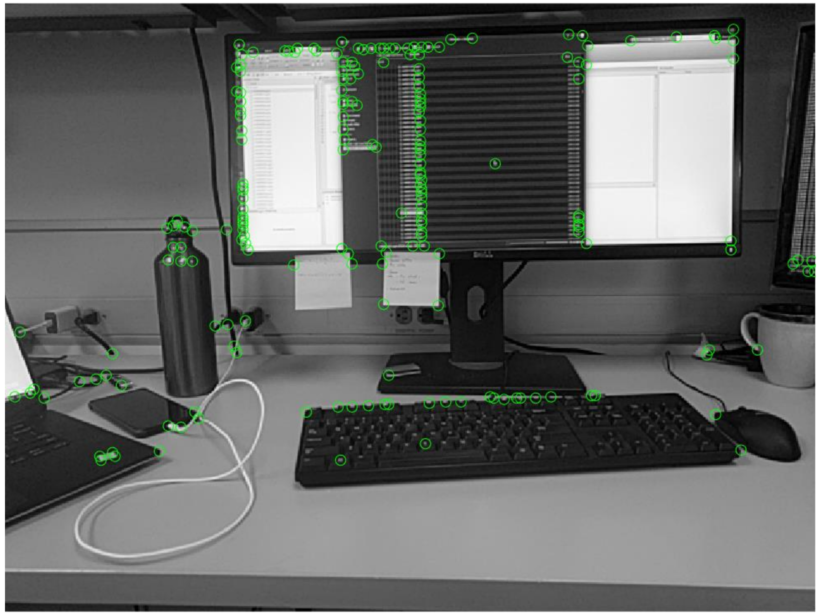
RS/TS effect



[1] F. M. Mirzaei and S. I. Roumeliotis, "**A Kalman Filter-based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation**," TRO'08
[2] C. Guo, D. G. Kottas, R. DuToit, A. Ahmed, R. Li, and S. I. Roumeliotis, "**Efficient visual-inertial navigation using a rolling-shutter camera with inaccurate timestamps**," RSS'14

Feature Extraction & Tracking

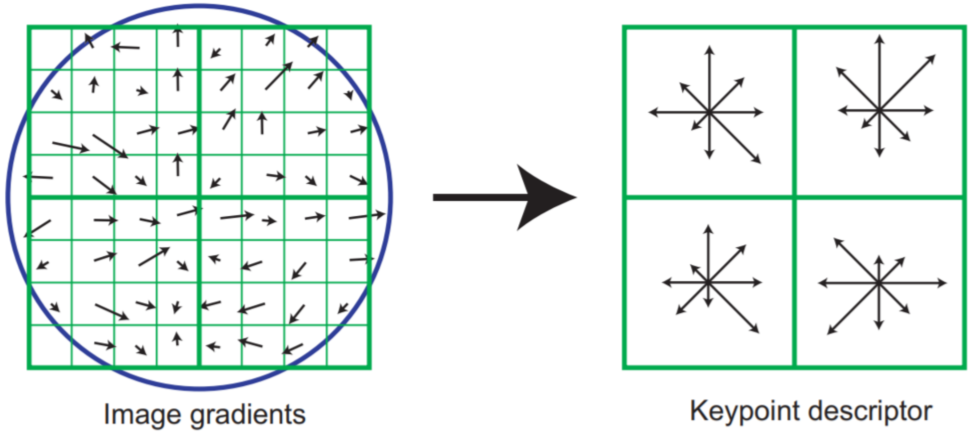
- Keypoint detection
  - Harris [1], DoG, FAST [2]

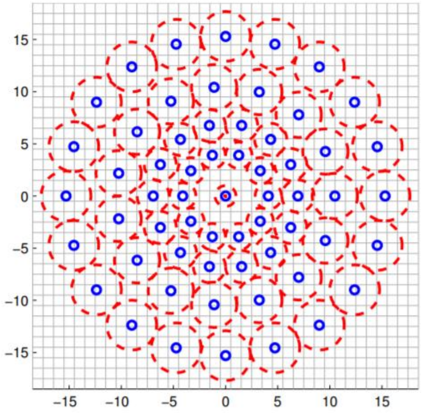


[1] C. Harris and M. Stephens, “**A Combined Corner and Edge Detector**,” Alvey Vision Conference’81
[2] R. Edward and T. Drummond, “**Machine learning for high-speed corner detection**,” ECCV’06

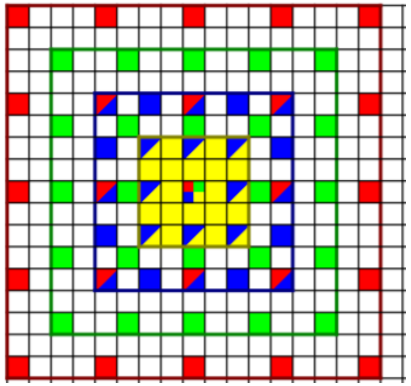
Feature Extraction & Tracking

- Keypoint detection
  - Harris [1], DoG, FAST [2]
- Descriptor extraction
  - SIFT[3], SURF[4], ORB[5], FREAK[6], BRISK[7], SDC[8]



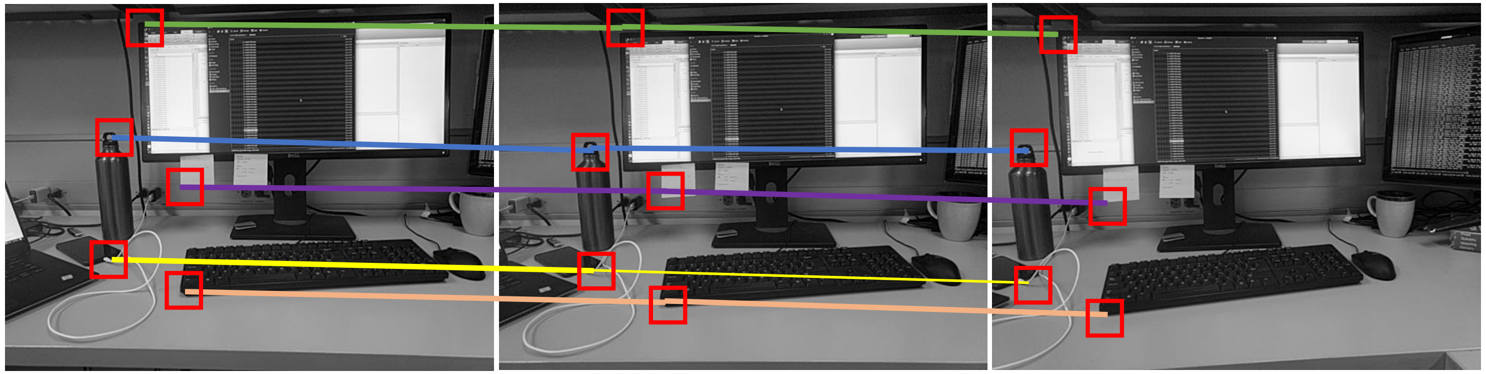


BRISK


SDC

[3] D. Lowe, “**Distinctive Image Features from Scale-Invariant Keypoints**,” IJCV’04
[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “**Speeded-up robust features (SURF)**,” Computer Vision and Image Understanding’08
[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “**ORB: An efficient alternative to SIFT or SURF**,” ICCV’11
[6] A. Alahi, R. Ortiz, Raphael, and P. Vandergheynst, “**FREAK: Fast Retina Keypoint**,” CVPR’12
[7] S. Leutenegger, M. Chli, and R. Siegwart, “**BRISK: Binary robust invariant scalable keypoints**,” ICCV’11
[8] R. Schuster, O. Wasenmuller, C. Unger, and D. Stricker, “**SDC – Stacked Dilated Convolution: A Unified Descriptor Network for Dense Matching,**” CVPR’19

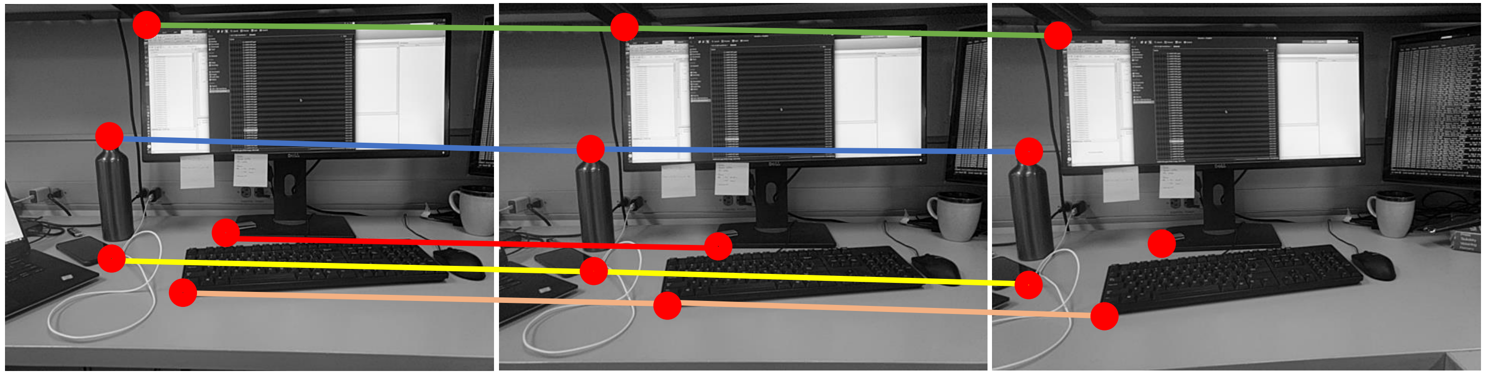
Feature Extraction & Tracking



- 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]

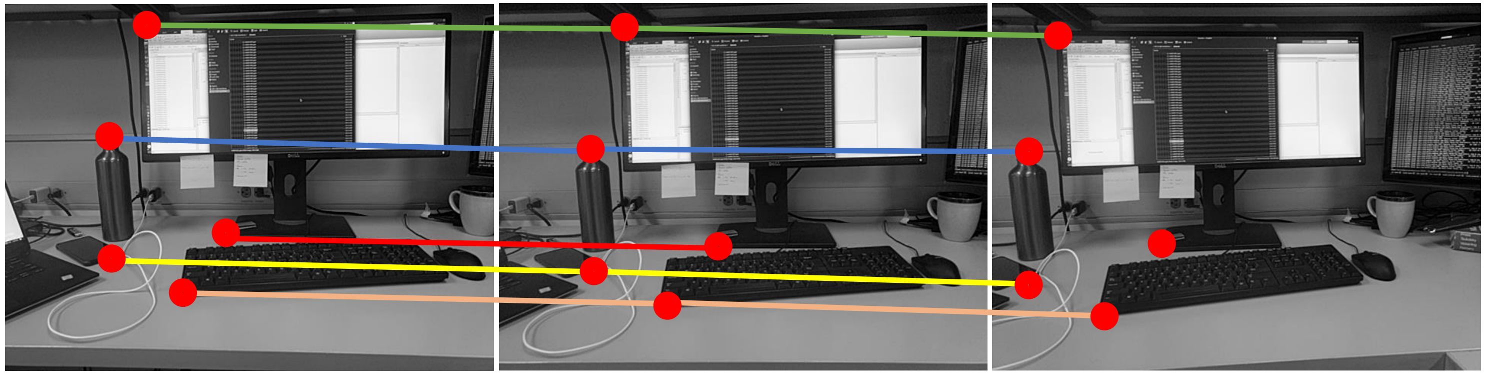
[9] B. Lucas and T. Kanade, “**An iterative image registration technique with an application to stereo vision**,” International Joint Conference on Artificaial Intelligence’88

Feature Extraction & Tracking

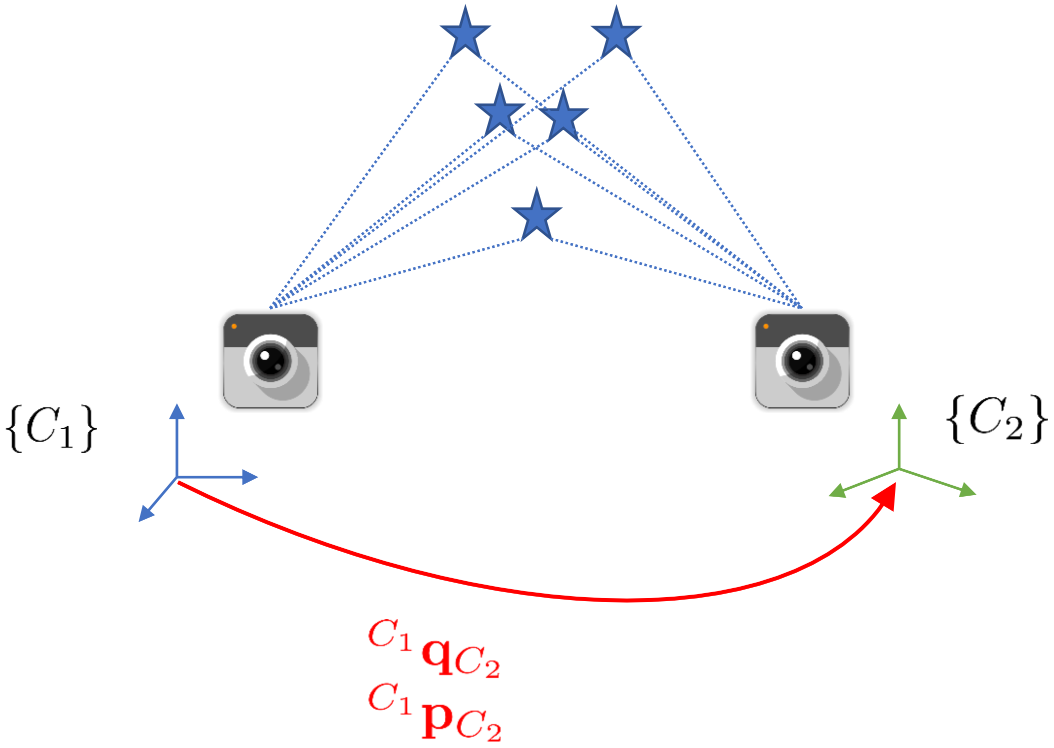


- Keypoint detection
  - Harris [1], DoG, FAST [2]
- Descriptor extraction
  - SIFT[3], SURF[4], ORB[5], FREAK[6], BRISK[7], SDC[8]
- Feature tracking (2D-to-2D)
  - KLT [9]
  - Descriptor-to-descriptor matching

Feature Extraction & Tracking

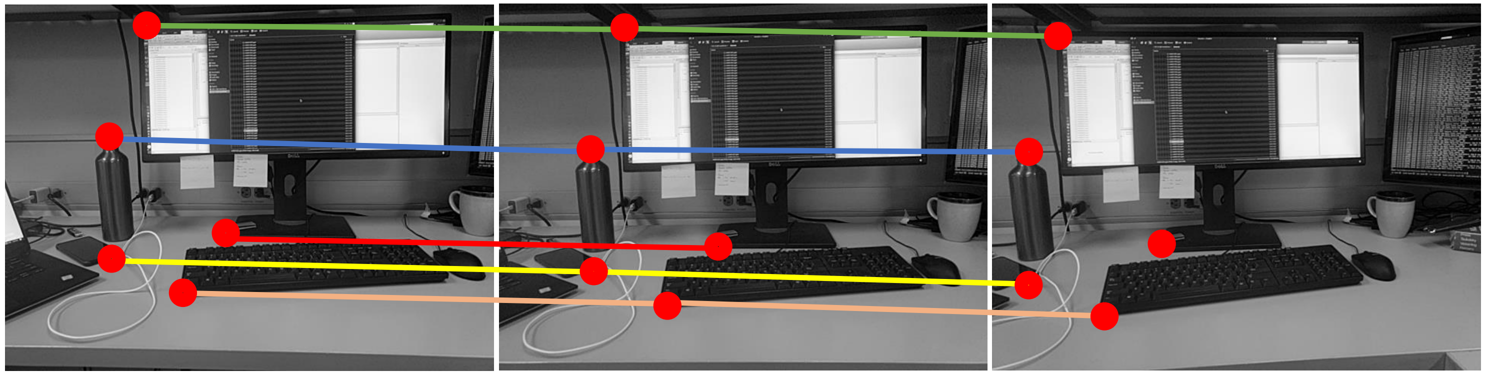


- Keypoint detection
  - Harris [1], DoG, FAST [2]
- Descriptor extraction
  - SIFT[3], SURF[4], ORB[5], FREAK[6], BRISK[7], SDC[8]
- Feature tracking (2D-to-2D)
  - KLT [9]
  - Descriptor-to-descriptor matching
  - Outlier rejection (RANSAC)
    - w/out gyro: 5pt RANSAC [10]

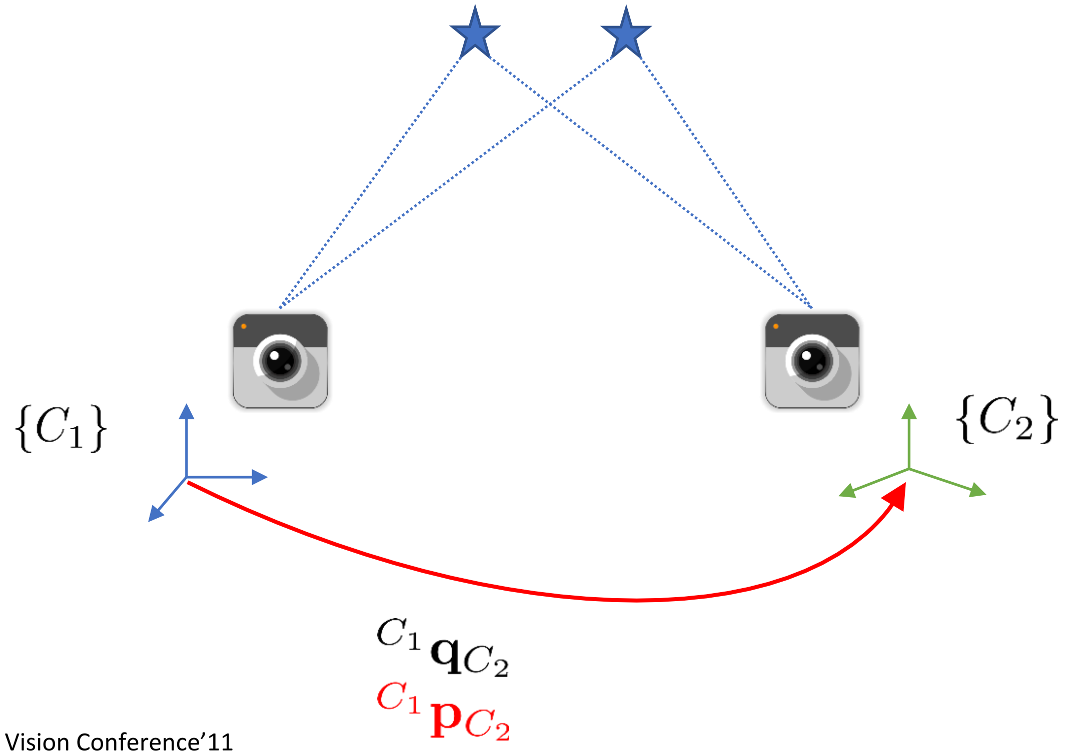


[10] D. Nister, **"An efficient solution to the five-point relative pose problem,"** TPAMI'04

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]
    - w/ gyro: 2pt RANSAC [11]



[10] D. Nister, **"An efficient solution to the five-point relative pose problem,"** TPAMI'04
[11] L. Kneip, M. Chli, and R. Siegwart, **"Robust real-time visual odometry with a single camera and an IMU,"** British Machine Vision Conference'11

Loop-closure Detection

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

VT

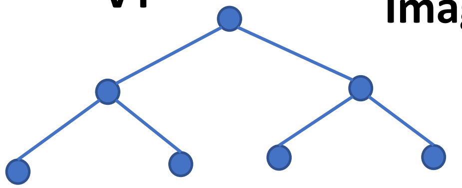
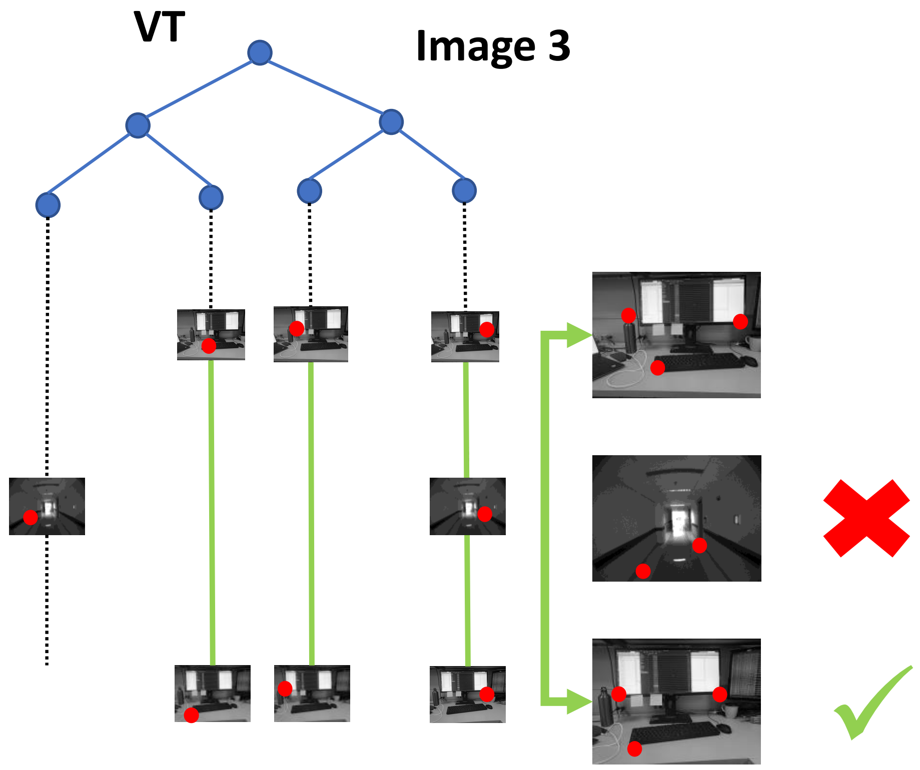


Image 1

[1] D. Nister and H. Stewenius, “**Scalable recognition with a vocabulary tree**,” CVPR’06

Loop-closure Detection

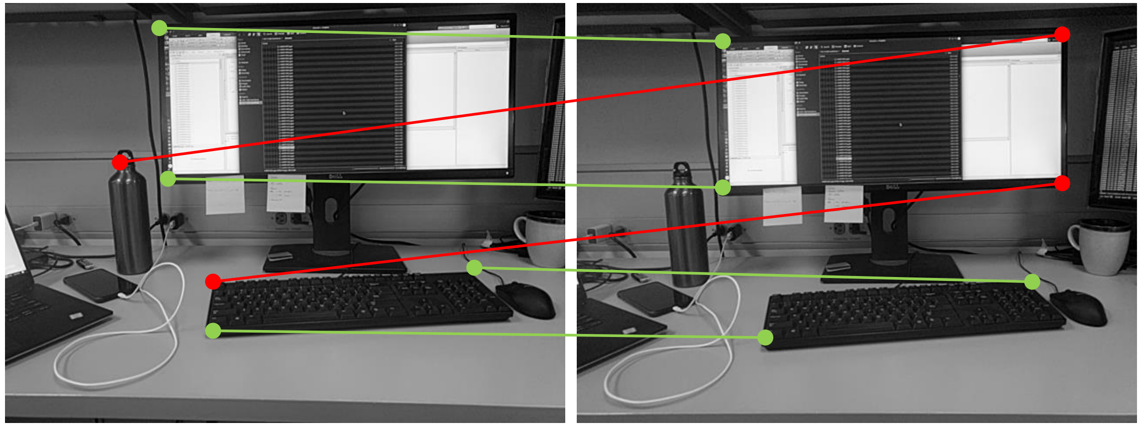
- Appearance-based image matching [1]
  - Create image descriptor from feature descriptors
  - Compare image descriptors against each other

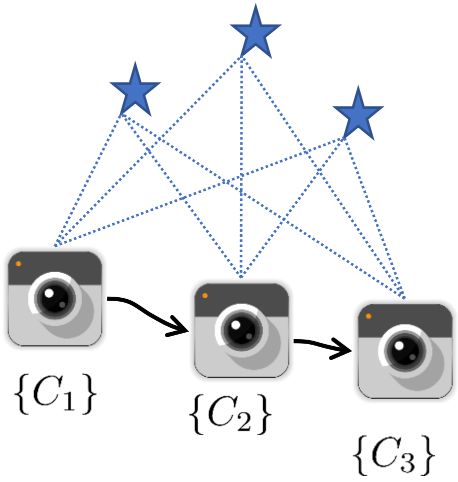


[1] D. Nister and H. Stewenius, "**Scalable recognition with a vocabulary tree**," CVPR'06

Loop-closure Detection

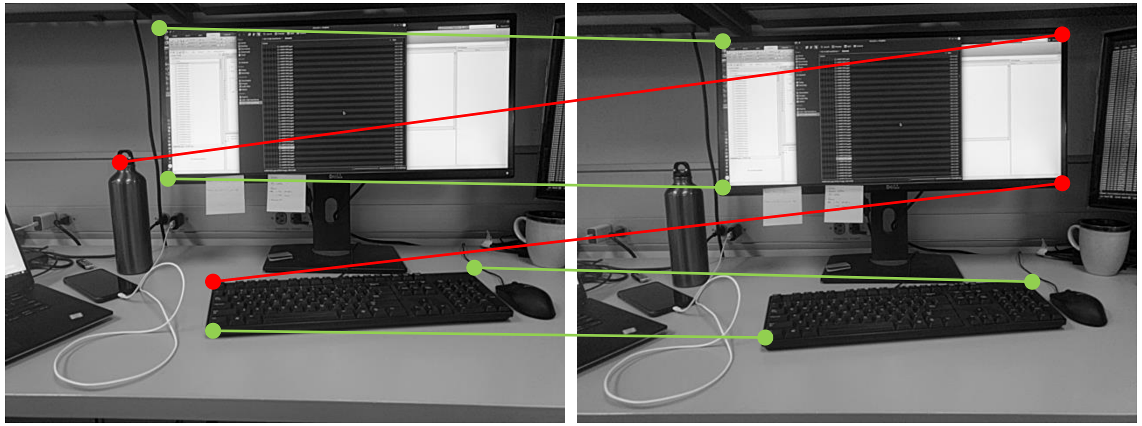
- Appearance-based image matching [1]
  - Create image descriptor from feature descriptors
  - Compare image descriptors against each other
- Outlier rejection / Geometric verification
  - 5pt[2] (3pt+1[3]) RANSAC to verify 2D-2D matches
  - P3P[4] (P2+1[5]) RANSAC for 2D-3D matches
  - Confirm loop-closure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





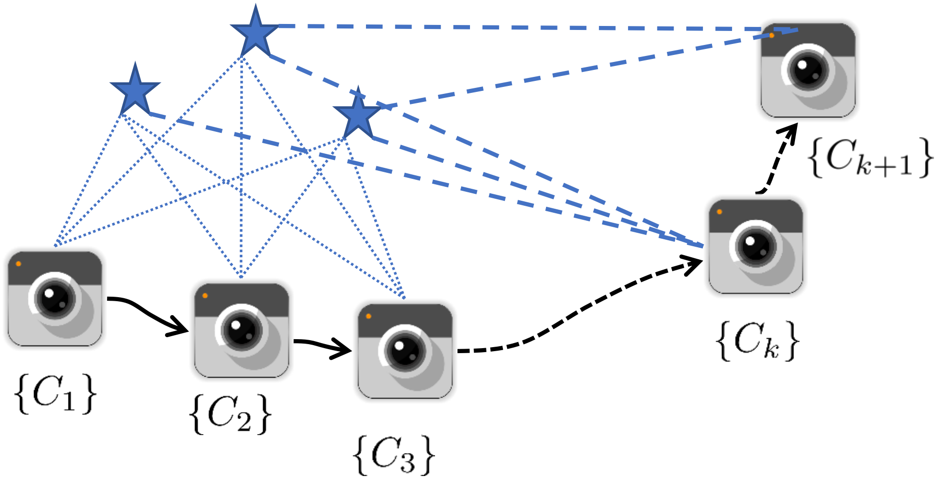
[2] D. Nister, **"An efficient solution to the five-point relative pose problem,"** TPAMI'04
[3] O. Naroditsky, X. Zhou, S. Roumeliotis, and K. Daniilidis, "**Two efficient solutions for visual odometry using directional correspondence**," TPAMI'12
[4] T. Ke, S. Roumeliotis, "**An Efficient Algebraic Solution to the Perspective-Three-Point Problem**," CVPR'17
[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[2] (3pt+1[3]) RANSAC to verify 2D-2D matches
  - P3P[4] (P2+1[5]) RANSAC for 2D-3D matches
  - Confirm loop-closure by matching consecutive images
    - Reduces false-positives
    - Delays map-based updates

LC



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

Sensor (IMU+Camera) Fusion

- Incremental BLS optimization[1]

$$\mathcal{C}_N = \sum_{k=1}^{N} \|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k} + \sum_{k=1}^{N} \sum_{j} \|\mathbf{z}_k^j - \mathbf{h}(\mathbf{x}_{I_k}, {}^G\mathbf{p}_{f_j})\|^2_{\sigma^2\mathbf{I}}$$

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

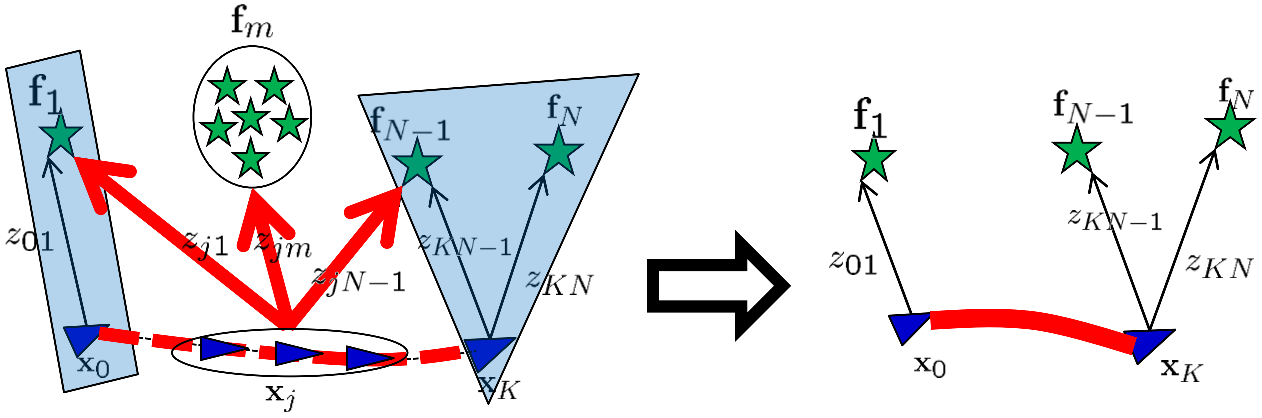
[1] M. Kaess, A. Ranganathan, and F. Dellaert, "**iSAM: Incremental smoothing and mapping**," TRO'08

Sensor (IMU+Camera) Fusion

- Incremental BLS optimization[1]

$$\mathcal{C}_N = \sum_{k=1}^{N} \|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k} + \sum_{k=1}^{N}\sum_{j} \|\mathbf{z}_k^j - \mathbf{h}(\mathbf{x}_{I_k}, {}^G\mathbf{p}_{f_j})\|^2_{\sigma^2\mathbf{I}}$$

  - Issue: Memory/CPU req.s increase w/ time
  - Remedy: C-KLAM[2] consistently marginalizes keyframes/features



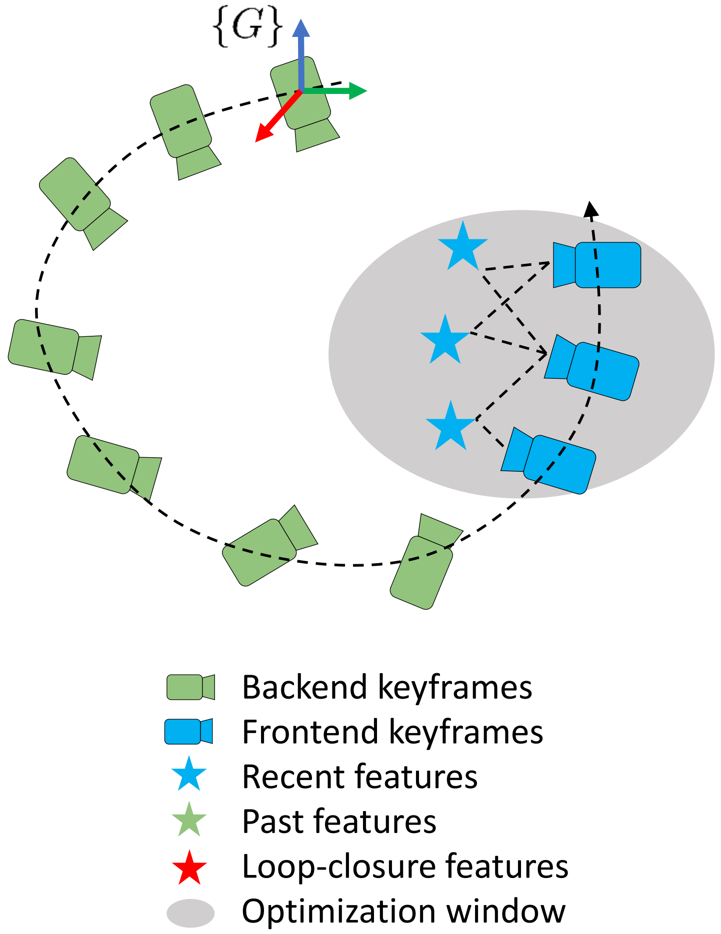
[2] E. Nerurkar, K. Wu, and S. Roumeliotis, "**C-KLAM: Constrained Keyframe-Based Localization and Mapping**," ICRA'14

Sensor (IMU+Camera) Fusion

- Incremental BLS optimization[1]

$$\mathcal{C}_N = \sum_{k=1}^{N} \|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k} + \sum_{k=1}^{N} \sum_{j} \|\mathbf{z}^j_k - \mathbf{h}(\mathbf{x}_{I_k}, {}^G\mathbf{p}_{f_j})\|^2_{\sigma^2\mathbf{I}}$$

  - 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)

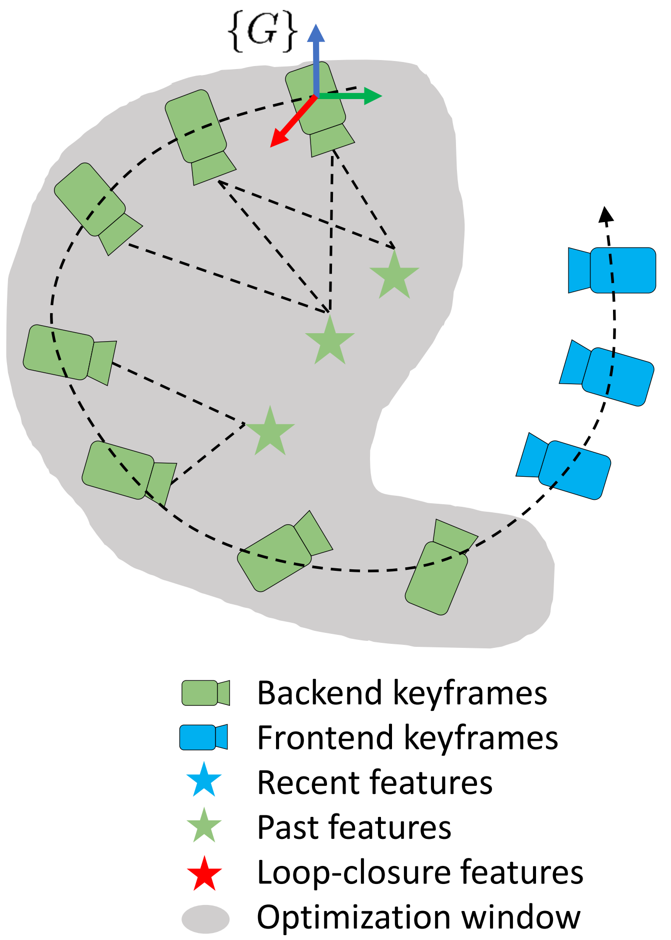


Sensor (IMU+Camera) Fusion

- Incremental BLS optimization[1]

$$\mathcal{C}_N = \sum_{k=1}^{N} \|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k} + \sum_{k=1}^{N}\sum_{j} \|\mathbf{z}_k^j - \mathbf{h}(\mathbf{x}_{I_k}, {}^G\mathbf{p}_{f_j})\|^2_{\sigma^2\mathbf{I}}$$

  - 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

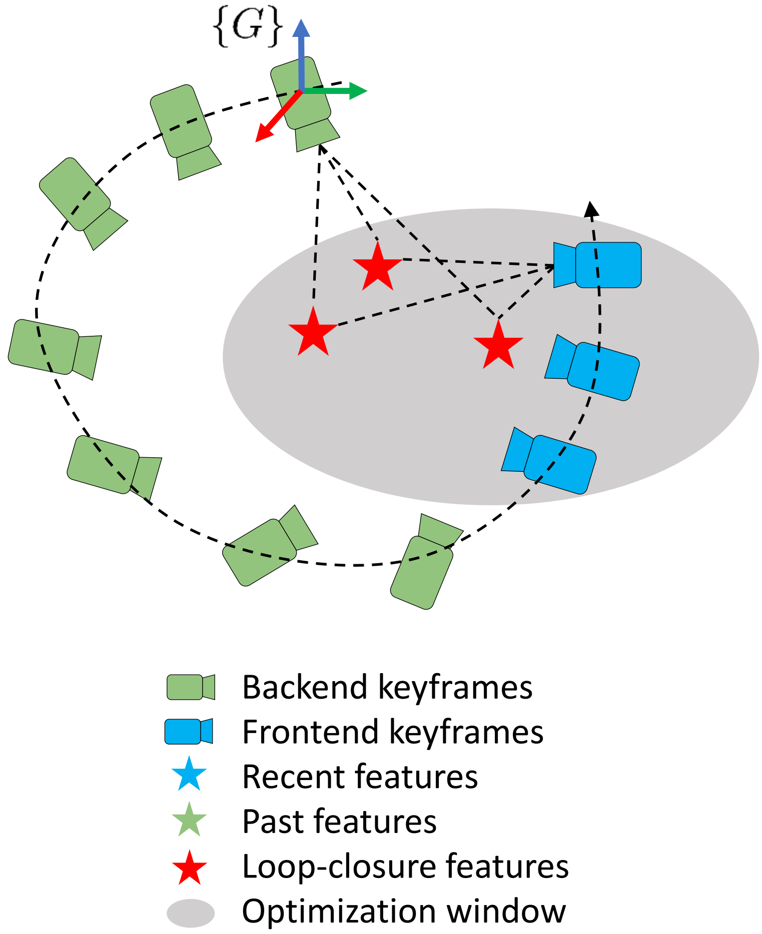


Sensor (IMU+Camera) Fusion

- Incremental BLS optimization[1]

$$\mathcal{C}_N = \sum_{k=1}^{N} \|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k} + \sum_{k=1}^{N}\sum_{j} \|\mathbf{z}_k^j - \mathbf{h}(\mathbf{x}_{I_k}, {}^G\mathbf{p}_{f_j})\|^2_{\sigma^2\mathbf{I}}$$

  - 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
  - Relocalize w/ loop closures
    - Assumes keyframes of backend as perfectly known -> inconsistency
      - Estimated covariance < true covariance



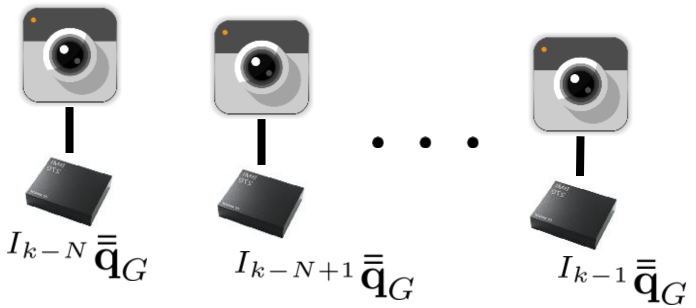
Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

- State Vector

$$\mathbf{x}_k = \left[\mathbf{x}_{I_k} \quad \mathbf{x}_{I_{k-1}} \quad {}^{I_{k-2}}\bar{\bar{\mathbf{q}}}_G \ldots {}^{I_{k-N}}\bar{\bar{\mathbf{q}}}_G \quad {}^{C}\bar{\bar{\mathbf{q}}}_I\right]^T$$

where $\mathbf{x}_I = \left[{}^{I}\mathbf{q}_G^T \quad \mathbf{b}_g^T \quad {}^{G}\mathbf{v}_I^T \quad \mathbf{b}_a^T \quad {}^{G}\mathbf{p}_I^T\right]^T$

$${}^{I}\bar{\bar{\mathbf{q}}}_G = \left[{}^{I}\mathbf{q}_G^T \quad {}^{G}\mathbf{p}_I^T\right]^T$$

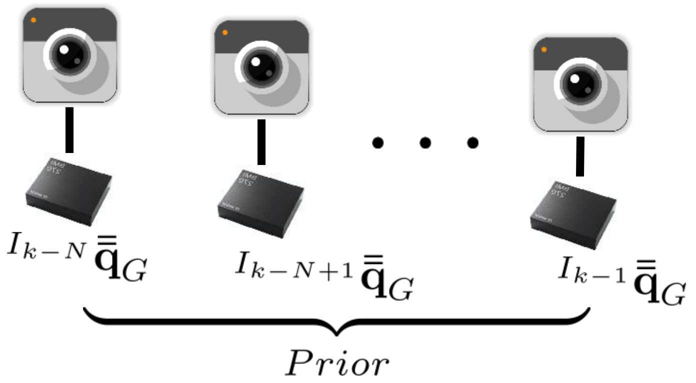


[1] A. Mourikis and S. Roumeliotis, “**A multi-state constraint Kalman filter for vision-aided inertial navigation**,” ICRA’07

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

- Step 1: Propagation

$$\mathcal{C}_{k|k-1} = \underbrace{\|\tilde{\mathbf{x}}_{k-1}\|^2_{\mathbf{P}_{k-1|k-1}}}_{Prior} + \underbrace{\|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k}}_{IMU}$$

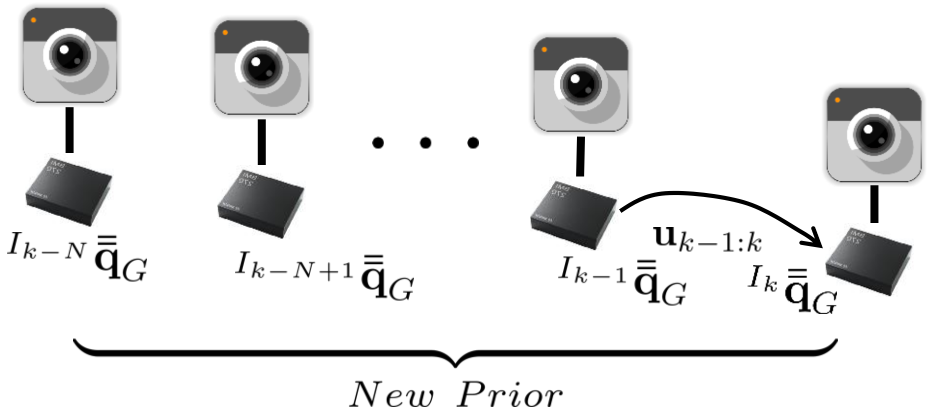


Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

- Step 1: Propagation

$$\mathcal{C}_{k|k-1} = \underbrace{\|\tilde{\mathbf{x}}_{k-1}\|^2_{\mathbf{P}_{k-1|k-1}}}_{Prior} + \underbrace{\|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k}}_{IMU}$$

$$\mathcal{C}_{k|k-1} \simeq \underbrace{\|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}}}_{New\ Prior}$$

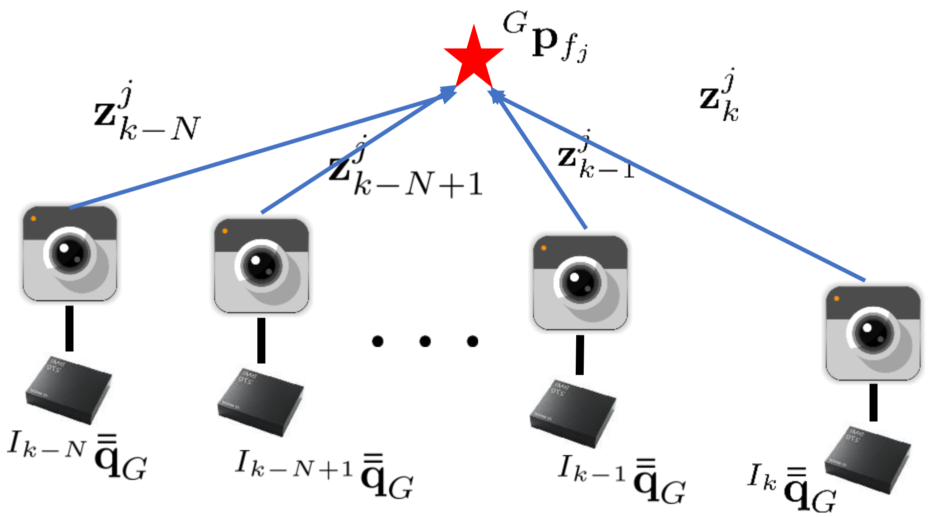


Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

- Step 1: Propagation

$$\mathcal{C}_{k|k-1} = \underbrace{\|\tilde{\mathbf{x}}_{k-1}\|^2_{\mathbf{P}_{k-1|k-1}}}_{Prior} + \underbrace{\|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k}}_{IMU}$$

$$\mathcal{C}_{k|k-1} \simeq \underbrace{\|\tilde{\mathbf{x}}_{k}\|^2_{\mathbf{P}_{k|k-1}}}_{New\ \ Prior}$$



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

$$\begin{aligned}
\mathcal{C}_{k|k} &= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \sum_j \|\mathbf{z}^j_{k-N:k} - \mathbf{f}(\mathbf{x}_{k-N:k}, \mathbf{p}_{f_j})\|^2 \\
&\simeq \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \sum_j \|\widetilde{\mathbf{z}}^j_{k-N:k} - (\mathbf{H}^j_x \widetilde{\mathbf{x}}_{k-N:k} + \mathbf{H}^j_f \widetilde{\mathbf{p}}_{f_j})\|^2 \\
&= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \sum_j \left(\|\mathbf{U}_j^T \widetilde{\mathbf{z}}^j_{k-N:k} - \mathbf{U}_j^T \mathbf{H}^j_x \widetilde{\mathbf{x}}_{k-N:k}\|^2 + \|\mathbf{V}_j^T \widetilde{\mathbf{z}}^j_{k-N:k} - (\mathbf{V}_j^T \mathbf{H}^j_x \widetilde{\mathbf{x}}_{k-N:k} + \mathbf{V}_j^T \mathbf{H}^j_f \widetilde{\mathbf{p}}_{f_j})\|^2 \right)
\end{aligned}$$

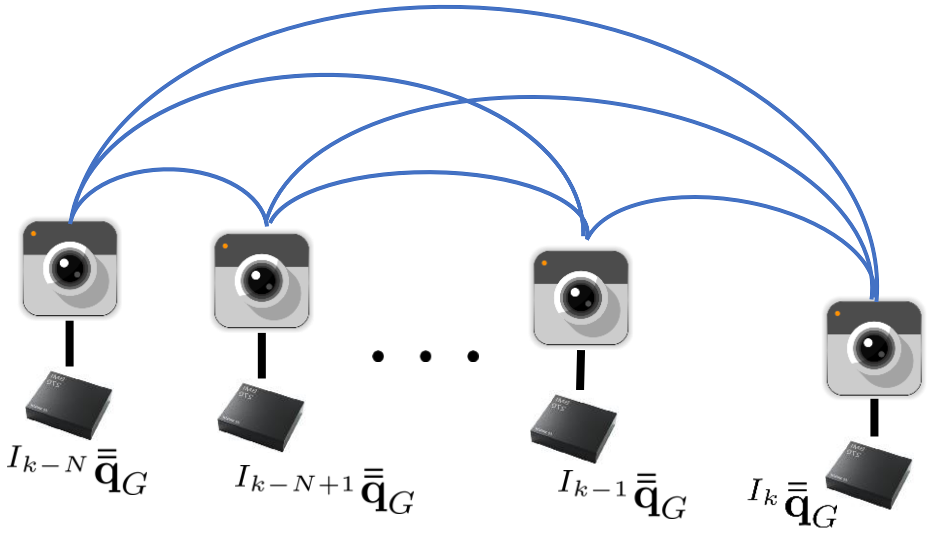
where $\begin{bmatrix} \mathbf{V}_j & \mathbf{U}_j \end{bmatrix}^T \begin{bmatrix} \mathbf{V}_j & \mathbf{U}_j \end{bmatrix} = \mathbf{I},\ \mathbf{U}_j^T \mathbf{H}^j_f = \mathbf{0}$

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

- Step 1: Propagation

$$\mathcal{C}_{k|k-1} = \underbrace{\|\tilde{\mathbf{x}}_{k-1}\|^2_{\mathbf{P}_{k-1|k-1}}}_{Prior} + \underbrace{\|\mathbf{x}_{I_k} - \mathbf{f}(\mathbf{x}_{I_{k-1}}, \mathbf{u}_{k-1:k})\|^2_{\mathbf{Q}_k}}_{IMU}$$

$$\mathcal{C}_{k|k-1} \simeq \underbrace{\|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}}}_{New\ \ Prior}$$



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

$$\begin{aligned}
\mathcal{C}_{k|k} &= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \sum_j \|\mathbf{z}^j_{k-N:k} - \mathbf{f}(\mathbf{x}_{k-N:k}, \mathbf{p}_{f_j})\|^2 \\
&\simeq \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \sum_j \|\widetilde{\mathbf{z}}^j_{k-N:k} - (\mathbf{H}^j_x \widetilde{\mathbf{x}}_{k-N:k} + \mathbf{H}^j_f \widetilde{\mathbf{p}}_{f_j})\|^2 \\
&= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \sum_j \left(\|\mathbf{U}^T_j \widetilde{\mathbf{z}}^j_{k-N:k} - \mathbf{U}^T_j \mathbf{H}^j_x \widetilde{\mathbf{x}}_{k-N:k}\|^2 + \|\mathbf{V}^T_j \widetilde{\mathbf{z}}^j_{k-N:k} - (\cancel{\mathbf{V}^T_j \mathbf{H}^j_x \widetilde{\mathbf{x}}_{k-N:k}} + \mathbf{V}^T_j \mathbf{H}^j_f \widetilde{\mathbf{p}}_{f_j})\|^2 \right)
\end{aligned}$$

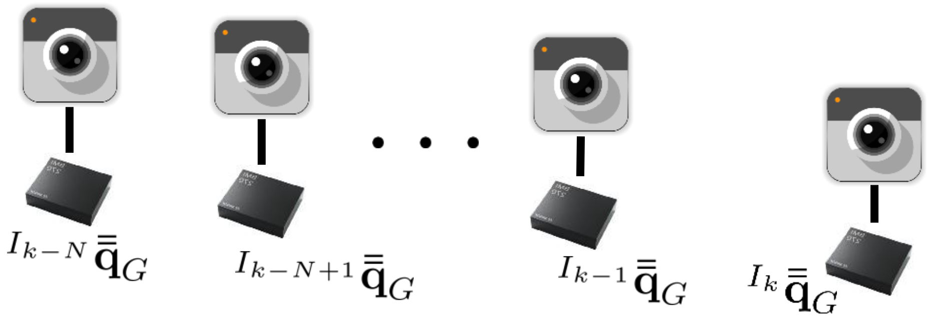
$$\mathcal{C}'_{k|k} = \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \sum_j \|\mathbf{U}^T_j \widetilde{\mathbf{z}}^j_{k-N:k} - \mathbf{U}^T_j \mathbf{H}^j_x \widetilde{\mathbf{x}}_{k-N:k}\|^2 \qquad \text{where } \begin{bmatrix}\mathbf{V}_j & \mathbf{U}_j\end{bmatrix}^T \begin{bmatrix}\mathbf{V}_j & \mathbf{U}_j\end{bmatrix} = \mathbf{I},\ \mathbf{U}^T_j \mathbf{H}^j_f = \mathbf{0}$$

$$= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \|\tilde{\mathbf{z}}_{k-N:k} - \mathbf{H}_x \widetilde{\mathbf{x}}_{k-N:k}\|^2$$

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

- Step 3: Update *[O(M³)]*

$$\begin{aligned}\mathcal{C}'_{k|k} &= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \|\tilde{\mathbf{z}}_{k-N:k} - \mathbf{H}_x \widetilde{\mathbf{x}}_{k-N:k}\|^2 \\ &= \|\tilde{\mathbf{x}}_{k-N:k}\|^2_{\mathbf{P}^{\oplus}_{k-N:k}}\end{aligned}$$



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

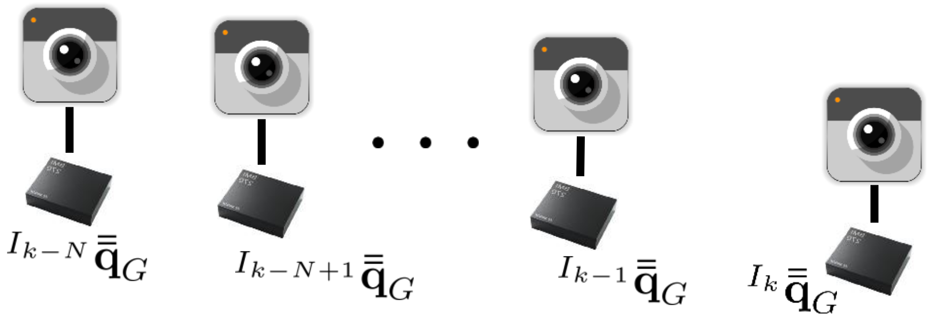
- Step 3: Update *[O(M³)]*

$$
\begin{aligned}
\mathcal{C}'_{k|k} &= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \|\tilde{\mathbf{z}}_{k-N:k} - \mathbf{H}_x \widetilde{\mathbf{x}}_{k-N:k}\|^2 \\
&= \|\tilde{\mathbf{x}}_{k-N:k}\|^2_{\mathbf{P}^{\oplus}_{k-N:k}}
\end{aligned}
$$

- Step 4: Marginalize the oldest pose ${}^{I_{k-N}}\bar{\bar{\mathbf{q}}}_G$

$$
\mathcal{C}'_{k|k} = \|\tilde{\mathbf{x}}_{k-N:k}\|^2_{\mathbf{P}^{\oplus}_{k-N:k}}
$$

$$
\mathcal{C}'_{k|k} = \|\tilde{\mathbf{x}}_{k-N+1:k}\|^2_{\mathbf{P}^{\oplus}_{k-N+1:k}} + \|\mathbf{L}_1 {}^{I_{k-N}}\tilde{\bar{\bar{\mathbf{q}}}}_G + \mathbf{L}_2 \tilde{\mathbf{x}}_{k-N+1:k}\|^2
$$



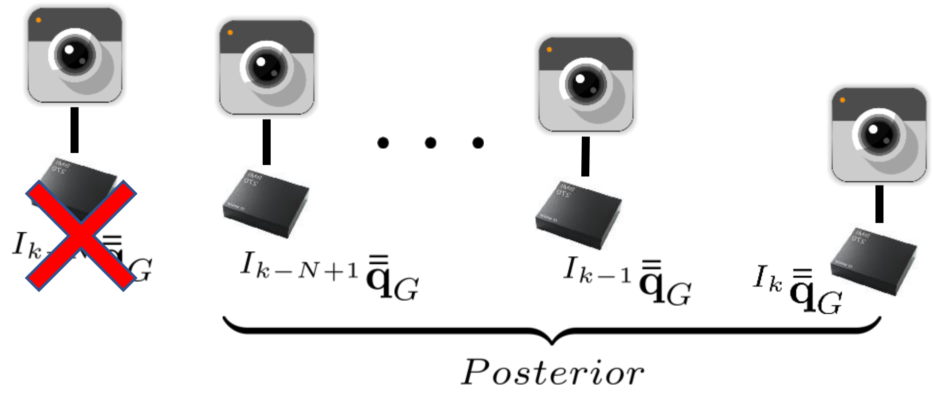
Frontend: Multi-state Constraint Kalman Filter (MSCKF) [1]

- Step 3: Update *[O(M³)]*

$$
\begin{aligned}
\mathcal{C}'_{k|k} &= \|\tilde{\mathbf{x}}_k\|^2_{\mathbf{P}_{k|k-1}} + \|\tilde{\mathbf{z}}_{k-N:k} - \mathbf{H}_x \tilde{\mathbf{x}}_{k-N:k}\|^2 \\
&= \|\tilde{\mathbf{x}}_{k-N:k}\|^2_{\mathbf{P}^{\oplus}_{k-N:k}}
\end{aligned}
$$

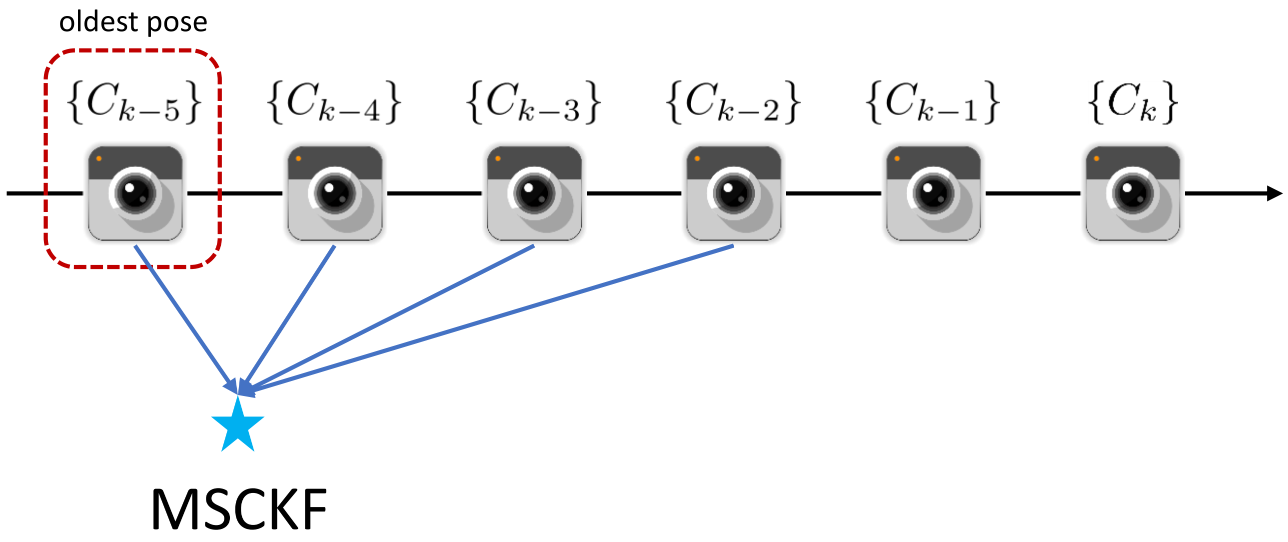
- Step 4: Marginalize the oldest pose ${}^{I_{k-N}}\bar{\bar{\mathbf{q}}}_G$

$$
\begin{aligned}
\mathcal{C}'_{k|k} &= \|\tilde{\mathbf{x}}_{k-N:k}\|^2_{\mathbf{P}^{\oplus}_{k-N:k}} \\
\mathcal{C}'_{k|k} &= \|\tilde{\mathbf{x}}_{k-N+1:k}\|^2_{\mathbf{P}^{\oplus}_{k-N+1:k}} + \|\mathbf{L}_1 {}^{I_{k-N}}\tilde{\bar{\bar{\mathbf{q}}}}_G + \mathbf{L}_2 \tilde{\mathbf{x}}_{k-N+1:k}\|^2 \\
\mathcal{C}''_{k|k} &= \underbrace{\|\tilde{\mathbf{x}}_{k-N+1:k}\|^2_{\mathbf{P}^{\oplus}_{k-N+1:k}}}_{Posterior}
\end{aligned}
$$



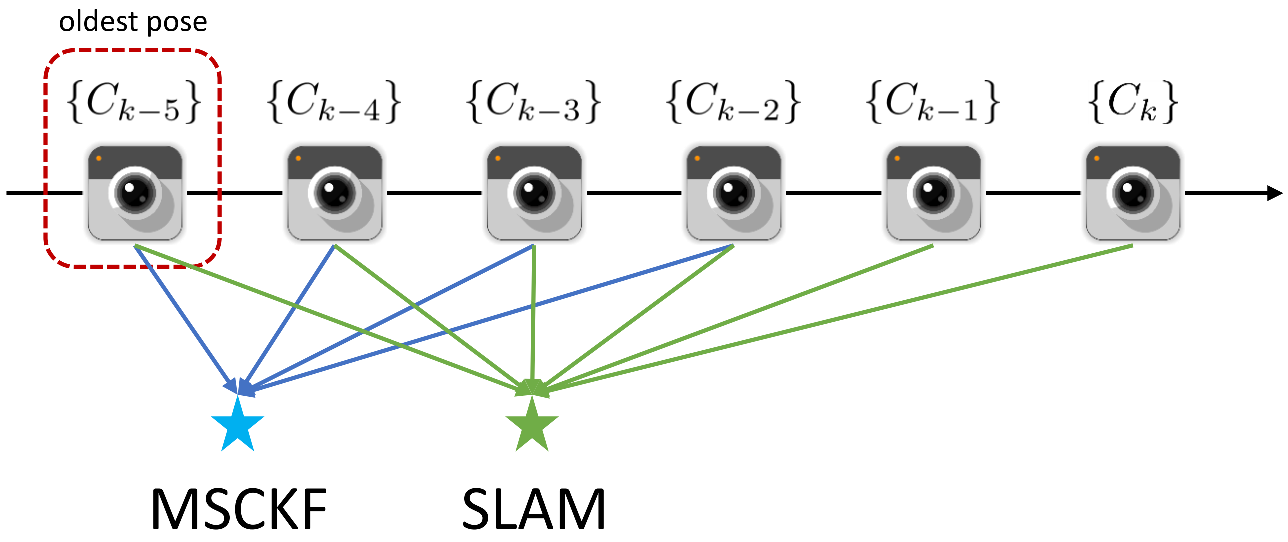
MSCKF Feature Classification & Processing

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



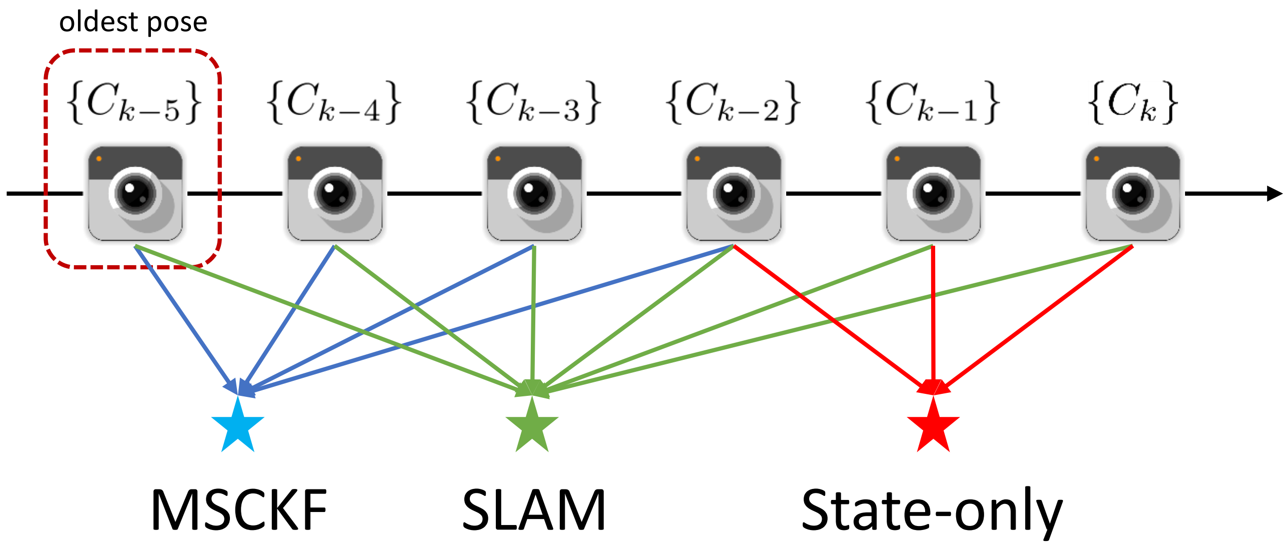
MSCKF Feature Classification & Processing

- 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) $\Longleftrightarrow$ MAP estimator w/ one Gauss-Newton iteration [1]
  - Iteratively processes camera meas/nts (Iterated EKF [2]), IMU meas/nts (IKS [3])
  - MSCKF (EKF) $\Longleftrightarrow$ 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)

	EKF	**EIF**
Prior	Covariance $\mathbf{P}$	Information $\mathbf{A}$
Propagation	$\mathbf{x} \leftarrow \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_\nu \end{bmatrix}$	
	$\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}\mathbf{\Phi}^T \\ \mathbf{\Phi}\mathbf{P} & \mathbf{\Phi}\mathbf{P}\mathbf{\Phi}^T + \mathbf{W} \end{bmatrix}$	$\mathbf{A} \leftarrow \begin{bmatrix} \mathbf{A} + \mathbf{\Phi}^T\mathbf{W}^{-1}\mathbf{\Phi} & -\mathbf{\Phi}^T\mathbf{W}^{-1} \\ -\mathbf{W}^{-1}\mathbf{\Phi} & \mathbf{W}^{-1} \end{bmatrix}$
Update	$\mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{\Sigma}$ $\mathbf{P}^{\oplus} = \mathbf{P} - \mathbf{P}\mathbf{H}^T\mathbf{S}^{-1}\mathbf{H}\mathbf{P}$ $\mathbf{P} \leftarrow \mathbf{P}^{\oplus}$ $\Delta\mathbf{x} = \mathbf{P}\mathbf{H}^T\mathbf{S}^{-1}\mathbf{r}$	$\mathbf{A}^{\oplus} = \mathbf{A} + \mathbf{H}^T\mathbf{\Sigma}^{-1}\mathbf{H}$ $\mathbf{b}^{\oplus} = \mathbf{H}^T\mathbf{\Sigma}^{-1}\mathbf{r}$ $\mathbf{A} \leftarrow \mathbf{A}^{\oplus}$ $\Delta\mathbf{x} = \mathbf{A}^{\oplus^{-1}}\mathbf{b}^{\oplus}$
	$\mathbf{x}^{\oplus} = \mathbf{x} + \Delta\mathbf{x}, \ \ \mathbf{x} \leftarrow \mathbf{x}^{\oplus}$	
Marginalization	$\mathbf{\Pi}\mathbf{x} = \begin{bmatrix} \mathbf{x}_\mu \\ \mathbf{x}_\rho \end{bmatrix}, \ \ \mathbf{x} \leftarrow \mathbf{x}_\rho$	
	$\mathbf{\Pi}\mathbf{P}\mathbf{\Pi}^T = \begin{bmatrix} \mathbf{P}_{\mu\mu} & \mathbf{P}_{\mu\rho} \\ \mathbf{P}_{\rho\mu} & \mathbf{P}_{\rho\rho} \end{bmatrix}$ $\mathbf{P} \leftarrow \mathbf{P}_{\rho\rho}$	$\mathbf{\Pi}\mathbf{A}\mathbf{\Pi}^T = \begin{bmatrix} \mathbf{A}_{\mu\mu} & \mathbf{A}_{\mu\rho} \\ \mathbf{A}_{\rho\mu} & \mathbf{A}_{\rho\rho} \end{bmatrix}$ $\mathbf{A}'_{\rho\rho} = \mathbf{A}_{\rho\rho} - \mathbf{A}_{\rho\mu}\mathbf{A}_{\mu\mu}^{-1}\mathbf{A}_{\mu\rho}$ $\mathbf{A} \leftarrow \mathbf{A}'_{\rho\rho}$

[1] A. H. Jazwinski, **Stochastic processes and filtering theory**, Academic Press, 1970
[2] P. S. Maybeck, **Stochastic Models, Estimation and Control**, vol. 1, Academic Press, 1979
[3] D. G. Kottas and S. I. Roumeliotis, "**An iterative Kalman smoother for robust 3D localization on mobile and wearable devices**," ICRA'15
[4] G. Sibley, L. Matthies, and G. Sukhatme, "**Sliding window filter with application to planetary landing**," JFR'10
[5] K. J. Wu, A. Ahmed, G. Georgiou, and S. I. Roumeliotis, "**A square root inverse filter for efficient vision-aided inertial navigation on mobile devices**," RSS'15

Inconsistency of VIO

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

[1] S. Julier and J. Uhlmann, "**A counter example to the theory of simultaneous localization and map building,**" ICRA'01
[2] J. A. Castellanos, J. Neira, and J. D. Tardos, "**Limits to the consistency of EKF-based slam**," IFAC'04
[3] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "**Observability-based rules for designing consistent EKF SLAM estimators,**" *IJRR'10*
[4] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "**Camera-IMU-based localization: Observability analysis and consistency improvement,**" *IJRR'*14

Inconsistency of VIO

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

Actual nonlinear system

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{l} \mathbf{f}_i(\mathbf{u}) \\ \mathbf{z} &= \mathbf{h}(\mathbf{x}) \end{cases}$$

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

$$\mathcal{O} = \begin{bmatrix} \nabla \mathfrak{L}^0 \mathbf{h} \\ \nabla \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h} \\ \nabla \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h} \\ \nabla \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h} \\ \vdots \end{bmatrix} = \boxed{\begin{bmatrix} \frac{\partial \mathfrak{L}^0 \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \vdots \end{bmatrix}} \boxed{\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}}} = \boxed{\boldsymbol{\Xi}} \boxed{\mathbf{B}}$$

- Finite. dim.
- Null(O) = Null(B)

Global trans.

$$\mathbf{Null}(\mathcal{O})_{VIO} = \begin{array}{c} \begin{bmatrix} \mathbf{0}_3 & {}^I_G\mathbf{C}\,{}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\lfloor \mathbf{v}\times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor \mathbf{p}\times \rfloor {}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_f\times \rfloor {}^G\mathbf{g} \end{bmatrix} \end{array} \begin{array}{c} \boldsymbol{\theta} \\ \mathbf{b_g} \\ \mathbf{v} \\ \mathbf{b_a} \\ \mathbf{p} \\ \mathbf{p}_f \end{array} = \begin{bmatrix} \mathbf{N}_t & | & \mathbf{N}_r \end{bmatrix}$$

Rot. around gravity

Ideal linearized system

$$\begin{cases} \mathbf{x}_{k+1|k} &= \boldsymbol{\Phi}_{k|k}\mathbf{x}_{k|k} + \mathbf{G}_{k|k}\mathbf{u}_k \\ \mathbf{z}_k &= \mathbf{H}_{k|k}\hat{\mathbf{x}}_{k|k} \end{cases}$$

[3] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "**Observability-based rules for designing consistent EKF SLAM estimators,**" *IJRR'10*

[4] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "**Camera-IMU-based localization: Observability analysis and consistency improvement,**" *IJRR'*14

Inconsistency of VIO

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

Actual nonlinear system

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{l} \mathbf{f}_i(\mathbf{u}) \\ \mathbf{z} &= \mathbf{h}(\mathbf{x}) \end{cases}$$

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

$$\mathcal{O} = \begin{bmatrix} \nabla \mathfrak{L}^0 \mathbf{h} \\ \nabla \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h} \\ \nabla \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h} \\ \nabla \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h} \\ \vdots \end{bmatrix} = \boxed{\begin{bmatrix} \frac{\partial \mathfrak{L}^0 \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \vdots \end{bmatrix}} \boxed{\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}}} = \boxed{\boldsymbol{\Xi}}\,\boxed{\mathbf{B}}$$

- Finite. dim.
- Null(O) = Null(B)

Global trans.

$$\mathbf{Null}(\mathcal{O})_{VIO} = \begin{bmatrix} \mathbf{0}_3 & {}^I_G\mathbf{C}\,{}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\lfloor \mathbf{v} \times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor \mathbf{p} \times \rfloor {}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_f \times \rfloor {}^G\mathbf{g} \end{bmatrix} \begin{matrix} \boldsymbol{\theta} \\ \mathbf{b_g} \\ \mathbf{v} \\ \mathbf{b_a} \\ \mathbf{p} \\ \mathbf{p}_f \end{matrix} = [\mathbf{N}_t \quad | \quad \mathbf{N}_r]$$

Rot. around gravity

Ideal linearized system

$$\begin{cases} \mathbf{x}_{k+1|k} &= \boldsymbol{\Phi}_{k|k}\mathbf{x}_{k|k} + \mathbf{G}_{k|k}\mathbf{u}_k \\ \mathbf{z}_k &= \mathbf{H}_{k|k}\hat{\mathbf{x}}_{k|k} \end{cases}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_k \\ \mathbf{H}_{k+1}\boldsymbol{\Phi}_{k|k} \\ \vdots \\ \mathbf{H}_{k+m}\boldsymbol{\Phi}_{k+m-1|k+m-1}\cdots\boldsymbol{\Phi}_{k|k} \end{bmatrix}$$

Actual linearized estimator

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k} &= \hat{\boldsymbol{\Phi}}_{k|k}\hat{\mathbf{x}}_{k|k} + \hat{\mathbf{G}}_{k|k}\mathbf{u}_k \\ \mathbf{z}_k &= \hat{\mathbf{H}}_{k|k}\hat{\mathbf{x}}_{k|k} \end{cases}$$

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{H}}_k \\ \hat{\mathbf{H}}_{k+1}\hat{\boldsymbol{\Phi}}_{k|k} \\ \vdots \\ \hat{\mathbf{H}}_{k+m}\hat{\boldsymbol{\Phi}}_{k+m-1|k+m-1}\cdots\hat{\boldsymbol{\Phi}}_{k|k} \end{bmatrix}$$

$$\boxed{\mathbf{Null}(\mathbf{M}) = \mathbf{Null}(\mathcal{O})_{VIO} \supsetneq \mathbf{Null}(\hat{\mathbf{M}}) = \mathbf{N}_t}$$

[3] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "**Observability-based rules for designing consistent EKF SLAM estimators,**" *IJRR'10*

[4] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "**Camera-IMU-based localization: Observability analysis and consistency improvement,**" *IJRR*'14

Inconsistency of VIO

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

Actual nonlinear system

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{l} \mathbf{f}_i(\mathbf{u}) \\ \mathbf{z} &= \mathbf{h}(\mathbf{x}) \end{cases}$$

$$\mathcal{O} = \begin{bmatrix} \nabla \mathfrak{L}^0 \mathbf{h} \\ \nabla \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h} \\ \nabla \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h} \\ \nabla \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h} \\ \vdots \end{bmatrix} = \boxed{\begin{bmatrix} \frac{\partial \mathfrak{L}^0 \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \vdots \end{bmatrix}} \boxed{\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}}} = \boxed{\boldsymbol{\Xi}} \boxed{\mathbf{B}}$$

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

- Finite. dim.
- Null(O) = Null(B)

Global trans.

$$\mathbf{Null}(\mathcal{O})_{VIO} = \begin{bmatrix} \mathbf{0}_3 & {}^I_G\mathbf{C}\,{}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\lfloor \mathbf{v} \times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor \mathbf{p} \times \rfloor {}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_f \times \rfloor {}^G\mathbf{g} \end{bmatrix} \begin{matrix} \boldsymbol{\theta} \\ \mathbf{b_g} \\ \mathbf{v} \\ \mathbf{b_a} \\ \mathbf{p} \\ \mathbf{p}_f \end{matrix} = \left[\mathbf{N}_t \quad | \quad \mathbf{N}_r\right]$$

Rot. around gravity

Ideal linearized system

$$\begin{cases} \mathbf{x}_{k+1|k} &= \boldsymbol{\Phi}_{k|k}\mathbf{x}_{k|k} + \mathbf{G}_{k|k}\mathbf{u}_k \\ \mathbf{z}_k &= \mathbf{H}_{k|k}\hat{\mathbf{x}}_{k|k} \end{cases}$$

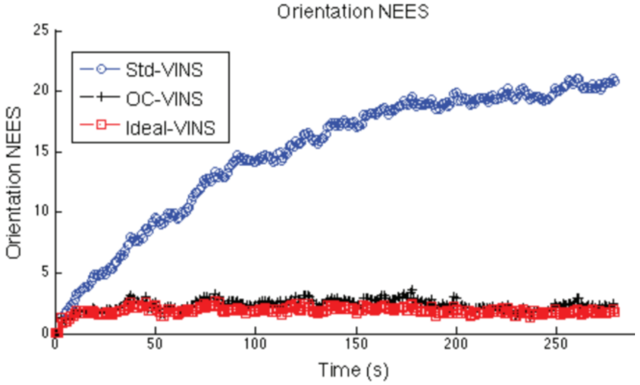
$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_k \\ \mathbf{H}_{k+1}\boldsymbol{\Phi}_{k|k} \\ \vdots \\ \mathbf{H}_{k+m}\boldsymbol{\Phi}_{k+m-1|k+m-1} \cdots \boldsymbol{\Phi}_{k|k} \end{bmatrix}$$

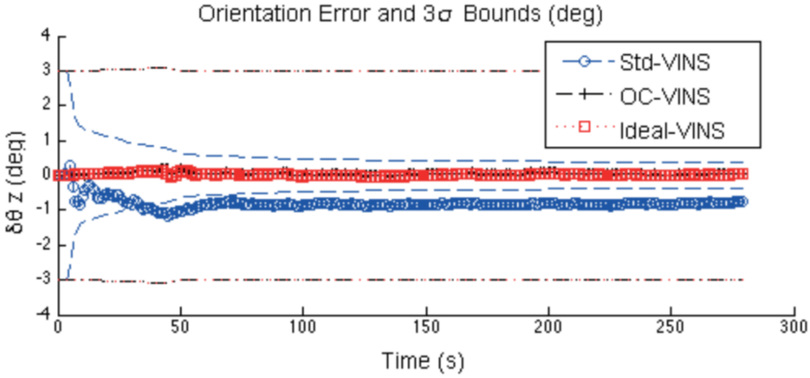
Actual linearized estimator

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k} &= \hat{\boldsymbol{\Phi}}_{k|k}\hat{\mathbf{x}}_{k|k} + \hat{\mathbf{G}}_{k|k}\mathbf{u}_k \\ \mathbf{z}_k &= \hat{\mathbf{H}}_{k|k}\hat{\mathbf{x}}_{k|k} \end{cases}$$

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{H}}_k \\ \hat{\mathbf{H}}_{k+1}\hat{\boldsymbol{\Phi}}_{k|k} \\ \vdots \\ \hat{\mathbf{H}}_{k+m}\hat{\boldsymbol{\Phi}}_{k+m-1|k+m-1} \cdots \hat{\boldsymbol{\Phi}}_{k|k} \end{bmatrix}$$

$$\boxed{\mathbf{Null}(\mathbf{M}) = \mathbf{Null}(\mathcal{O})_{VIO} \quad \supsetneq \quad \mathbf{Null}(\hat{\mathbf{M}}) \;=\; \mathbf{N}_t}$$



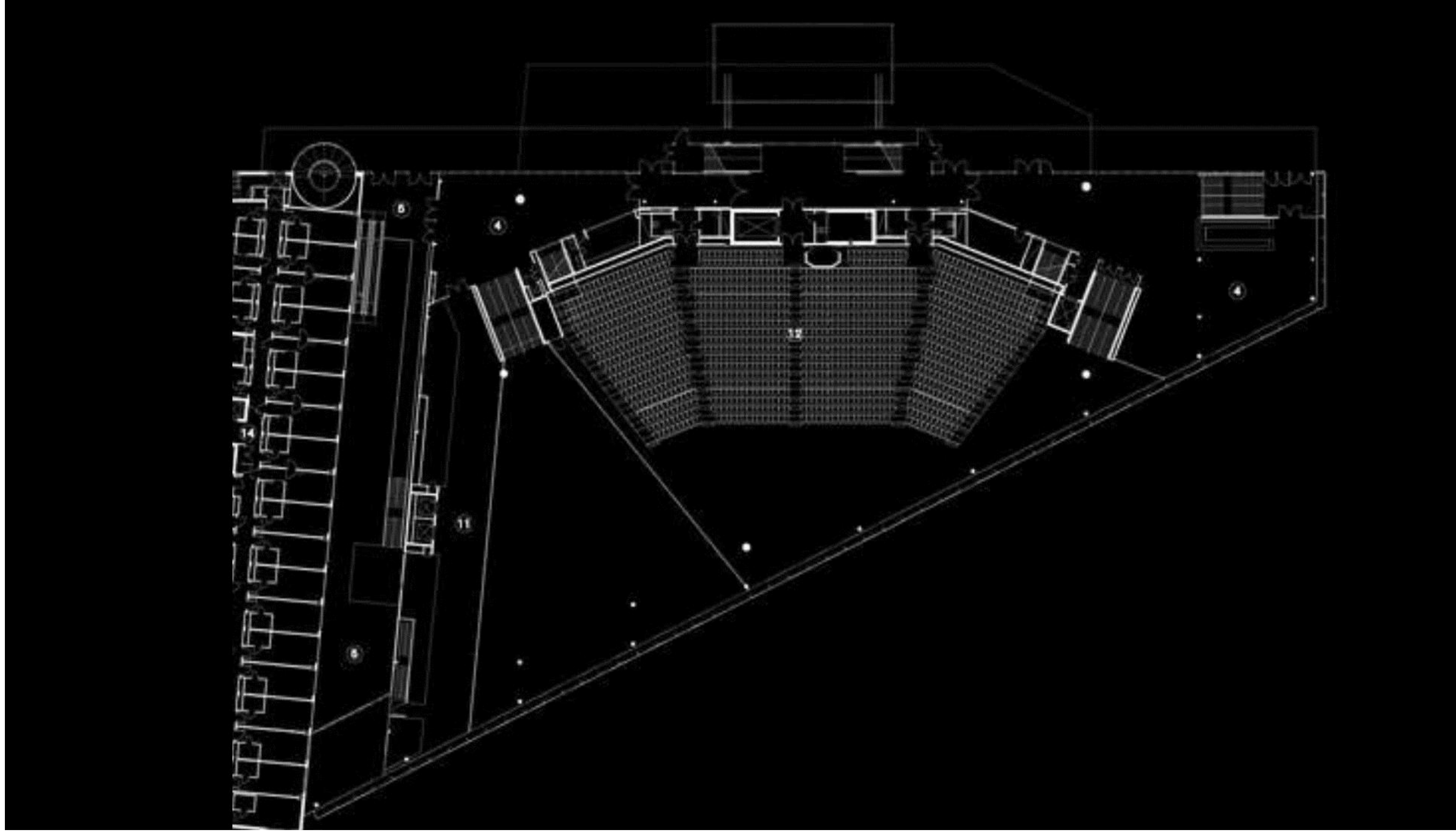


[3] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "**Observability-based rules for designing consistent EKF SLAM estimators,**" *IJRR'10*

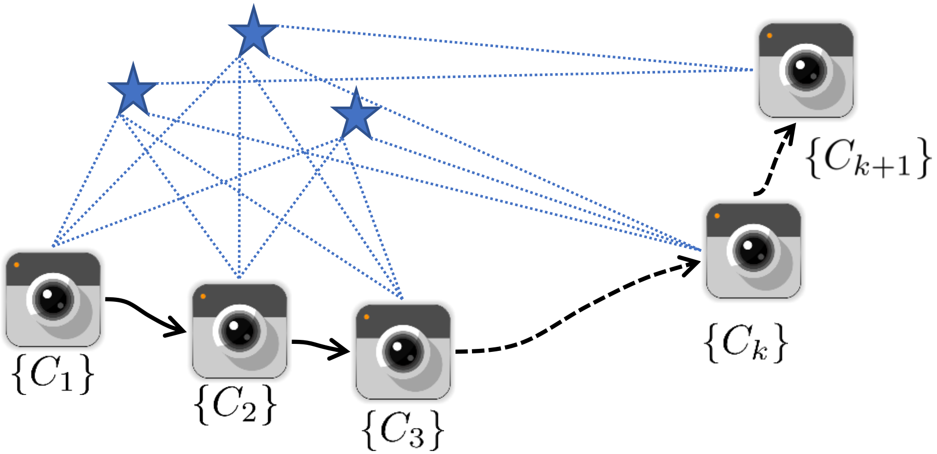
[4] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "**Camera-IMU-based localization: Observability analysis and consistency improvement,**" *IJRR'*14

Mapping Backend

- Offline: BA [1,2], CM [3]



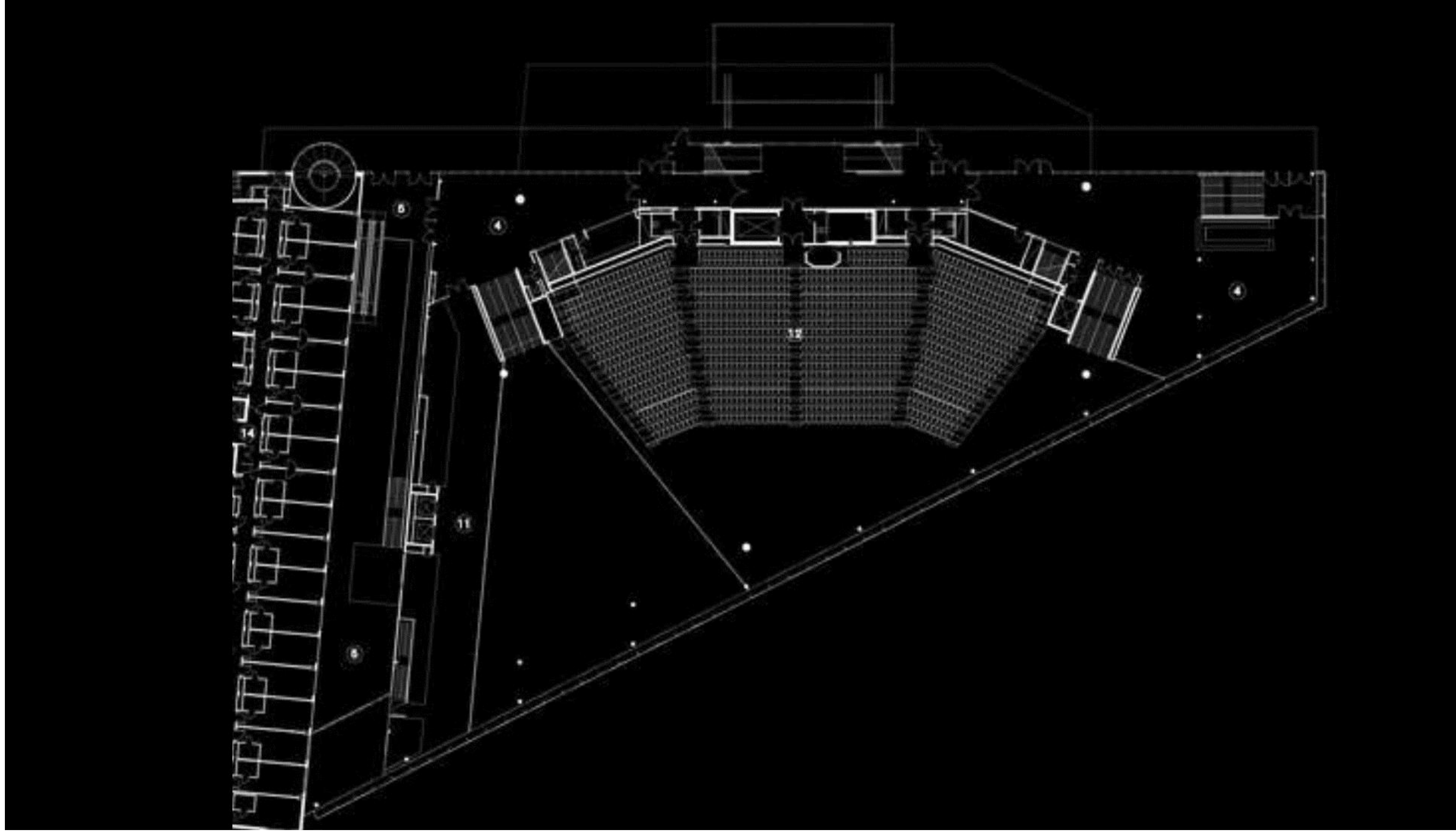
ICRA'16



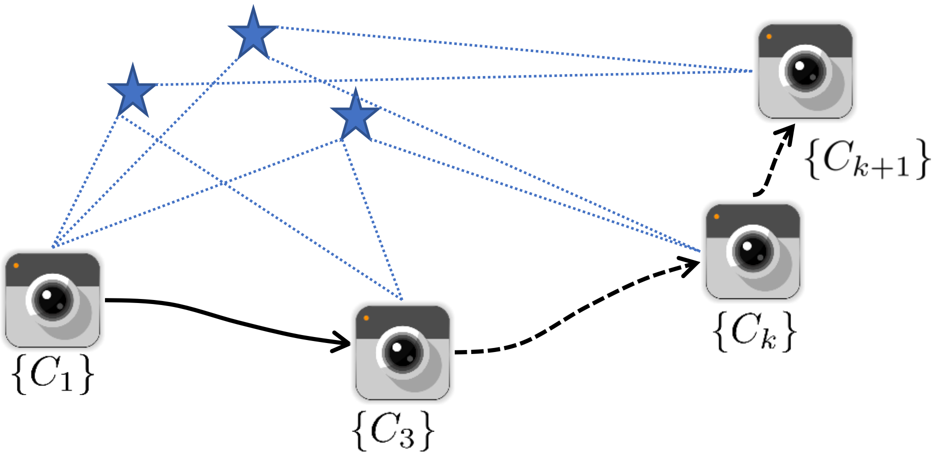
[1] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "**Bundle Adjustment - A Modern Synthesis**," Vision Algorithms: Theory and Practice, 2000
[2] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, R. Siegwart, "**Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization**," RSS'14
[3] C. Guo, K. Sartipi, R. DuToit, G. Georgiou, R. Li, J. O'Leary, E. Nerurkar, J. Hesch, and S. Roumeliotis, "**Resource-Aware Large-Scale Cooperative Three-Dimensional Mapping Using Multiple Mobile Devices**," TRO'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



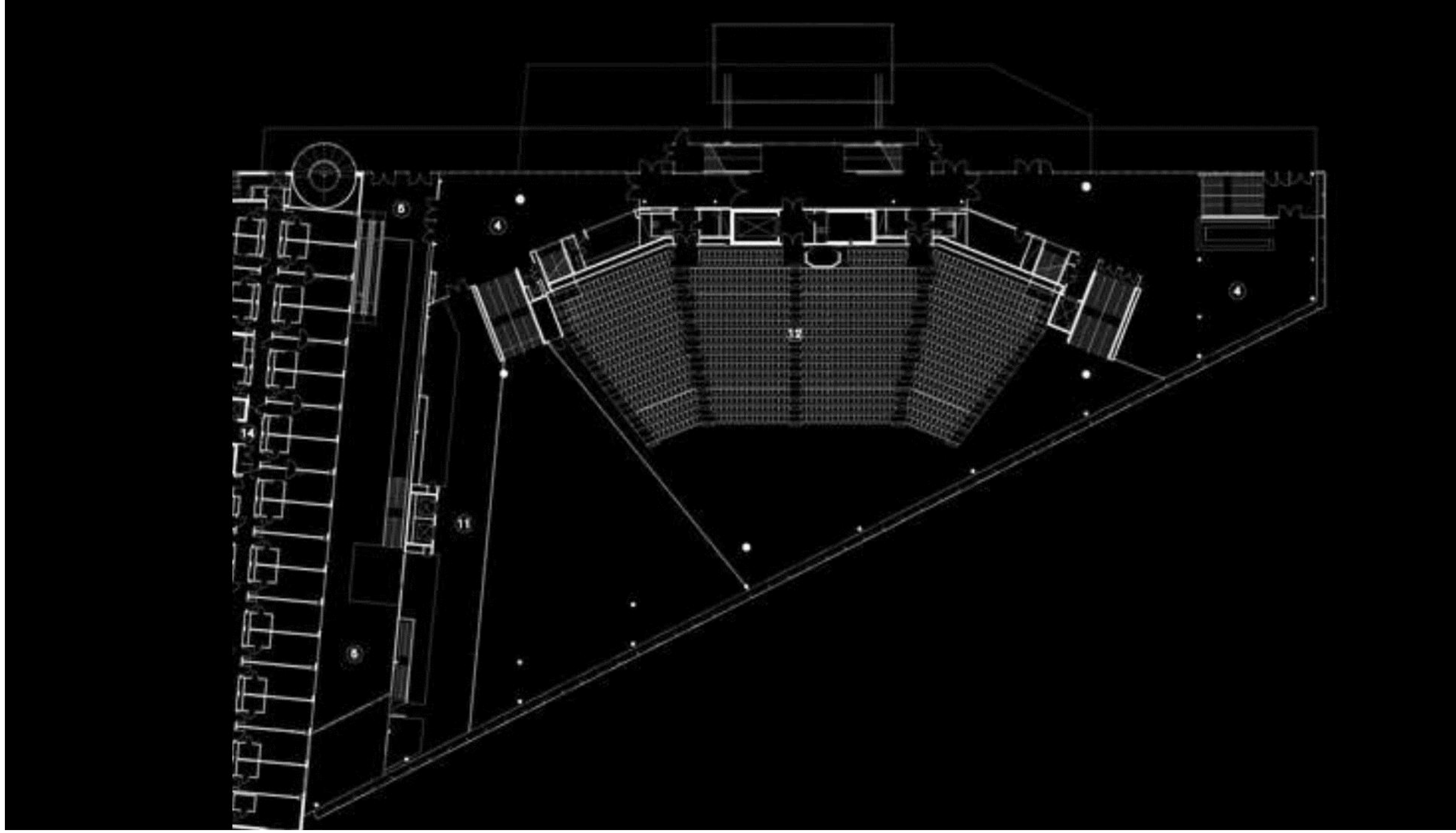
ICRA’16



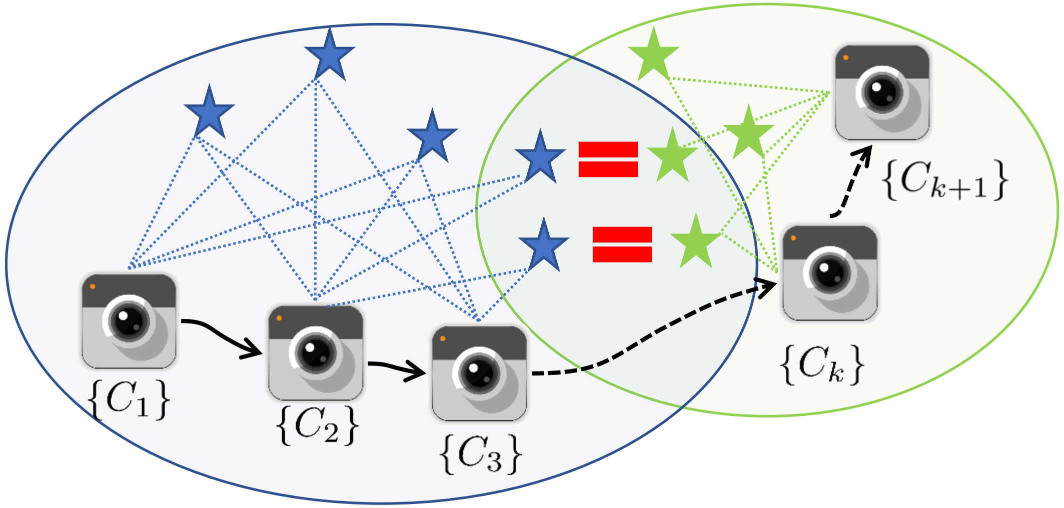
[4] G. Klein and D. Murray, “**Parallel Tracking and Mapping for Small AR Workspaces**,” ISMAR’07
[5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “**iSAM2: Incremental Smoothing and Mapping using the Bayes Tree**,” IJRR’12
[6] E.D. Nerurkar, K.J. Wu, and S.I. Roumeliotis, “**C-KLAM: Constrained Keyframe-Based Localization and Mapping**,” ICRA’14

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



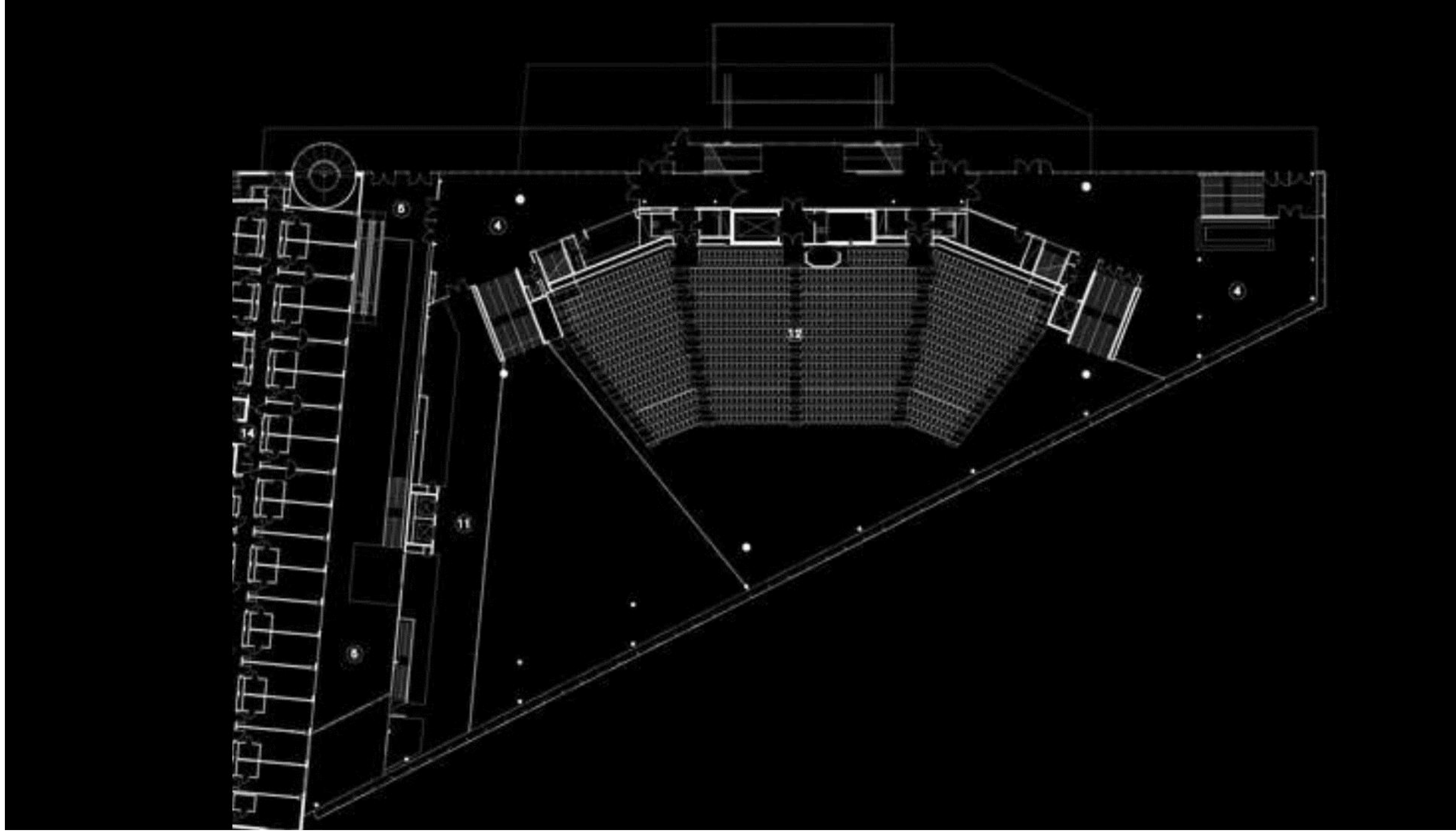
ICRA'16



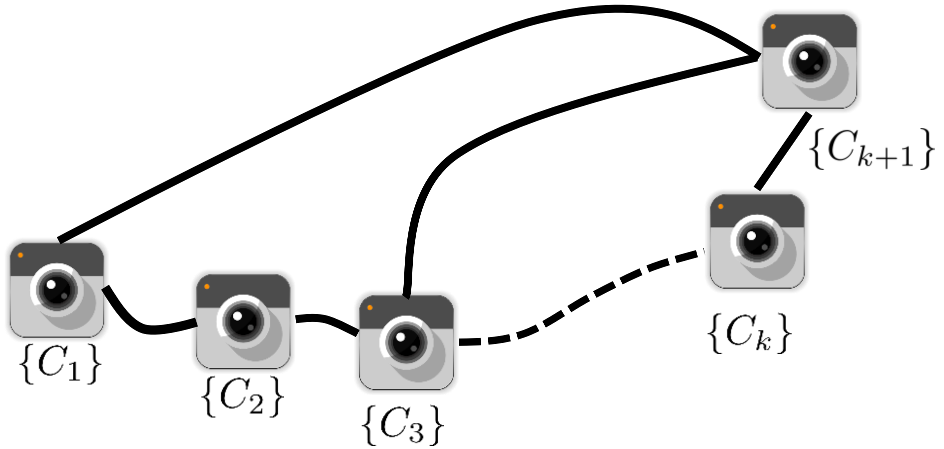
[7] K. Ni, D. Steedly, and F. Dellaert, "**Tectonic SAM: Exact, out-of-core, submap-based SLAM**," ICRA'07
[8] K. Sartipi and S. Roumeliotis, "**Efficient alignment of visual-inertial maps**," ISER'18

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



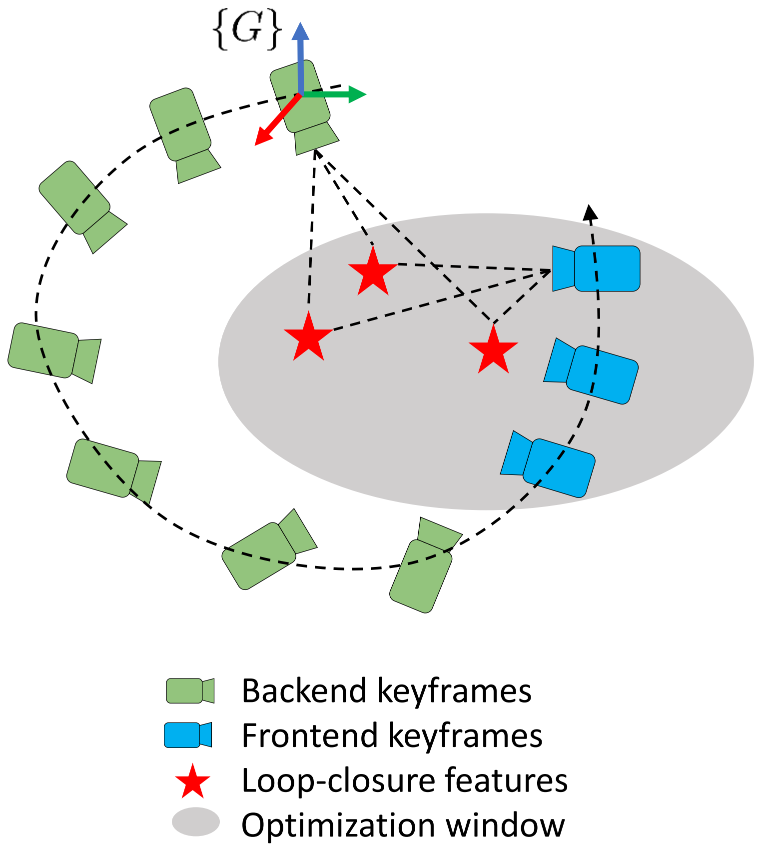
ICRA'16



[9] J. Gutmann and K. Konolige, "**Incremental Mapping of Large Cyclic Environments**," CIRA'99
[10] S. Thrun and M. Montemerlo, "**The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures**," IJRR'05
[11] T. Qin, P. Li, and S. Shen, "**VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator**," TRO'18

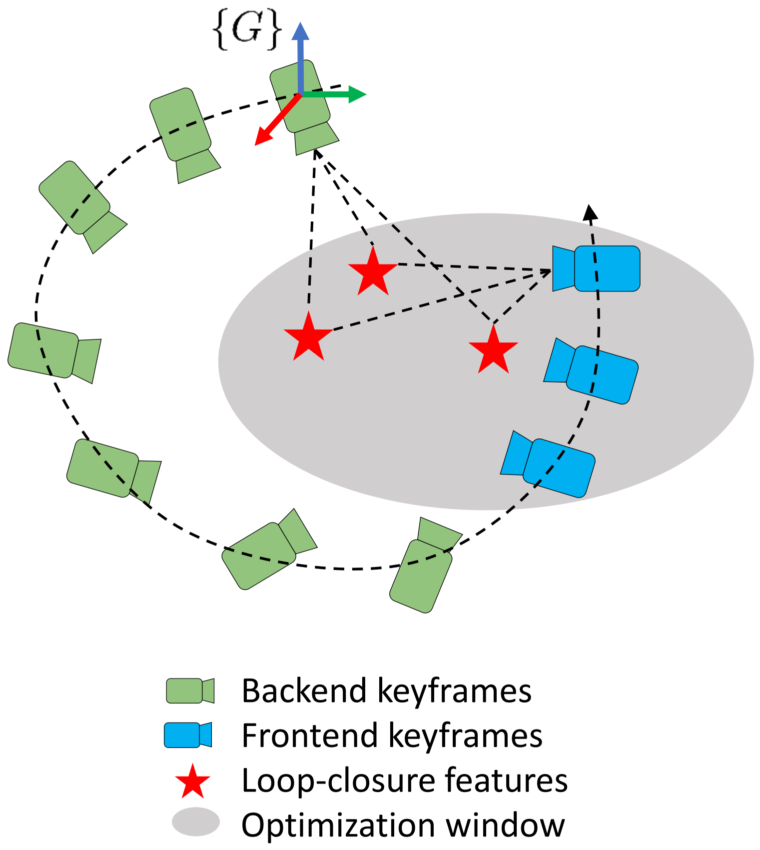
Map-based Updates

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



Map-based Updates

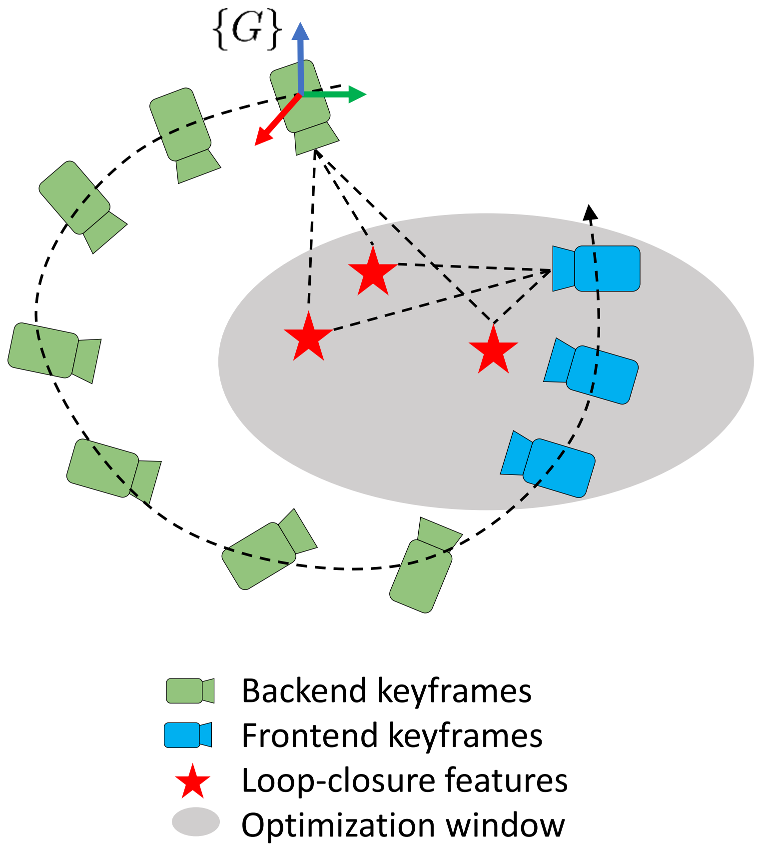
- Mapped keyframes/key features provided by the Backend to the Frontend
- Map assumed perfectly known [1,2]



[1] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "**Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing**," TRO'09
[2] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "**Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization**," RSS'14

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



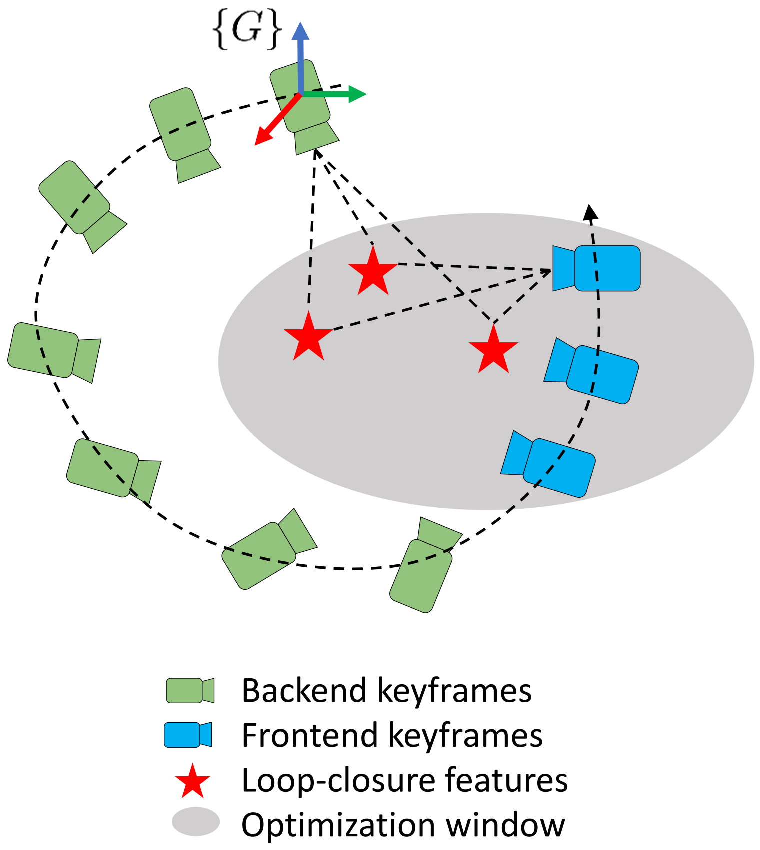
$$\mathcal{C}_{map} = ||\mathbf{z}_k^j - \mathbf{h}(\mathbf{x}_k, \mathbf{p}_{f_j})||^2_{\sigma^2\mathbf{I}} \simeq ||\tilde{\mathbf{z}}_k^j - \mathbf{H}_k^j \tilde{\mathbf{x}}_k - \cancel{\mathbf{H}_f^j \tilde{\mathbf{p}}}_{f_j}||^2_{\sigma^2\mathbf{I}}$$

[1] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "**Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing**," TRO'09
[2] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "**Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization**," RSS'14

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

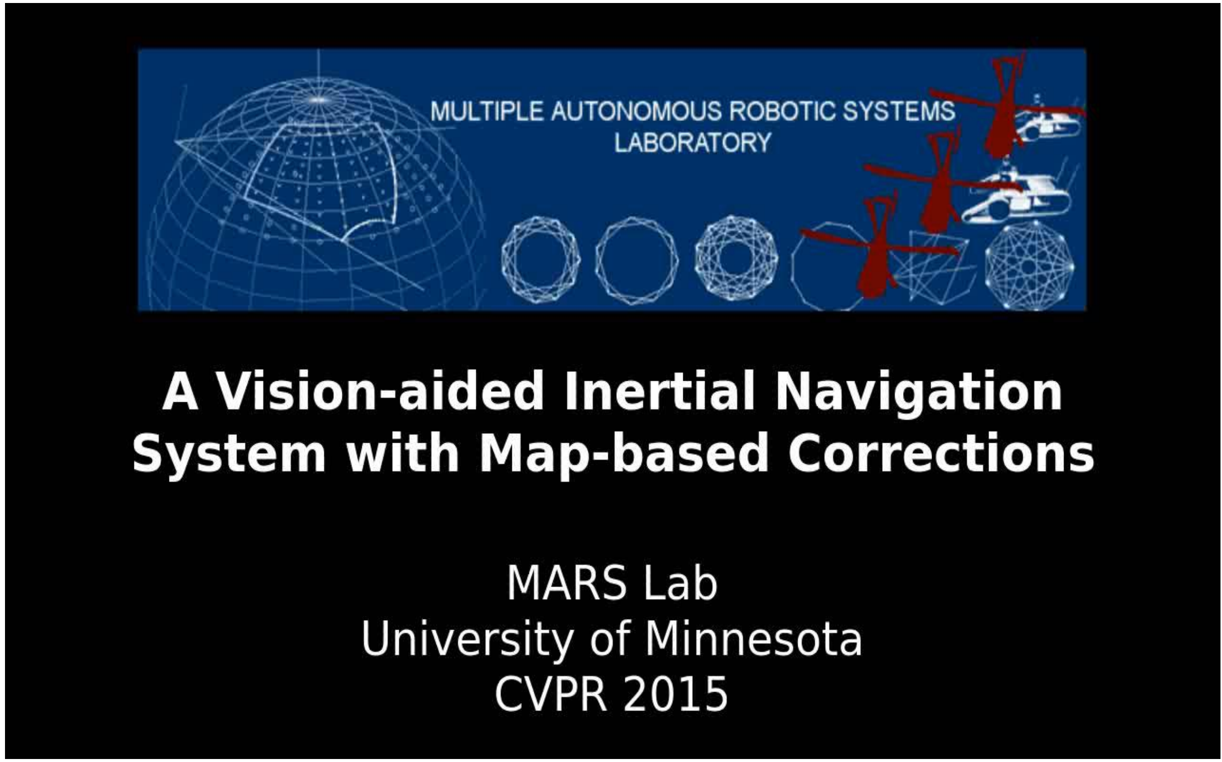
$$\mathcal{C}_{map} = ||\mathbf{z}_k^j - \mathbf{h}(\mathbf{x}_k, \mathbf{p}_{f_j})||^2_{\sigma^2\mathbf{I}} \simeq ||\tilde{\mathbf{z}}_k^j - \mathbf{H}_k^j\tilde{\mathbf{x}}_k - \cancel{\mathbf{H}_f^j\tilde{\mathbf{p}}}_{f_j}||^2_{4\sigma^2\mathbf{I}}$$



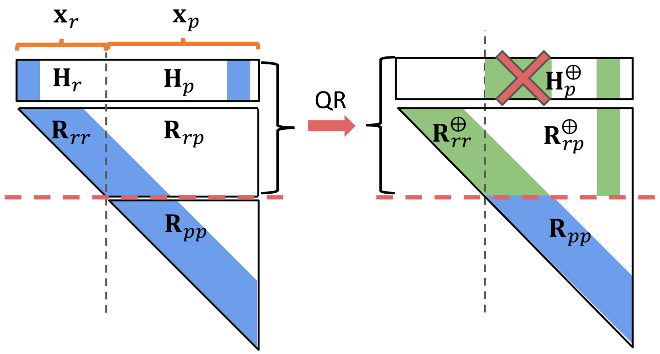
[1] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "**Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing**," TRO'09
[2] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "**Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization**," RSS'14

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



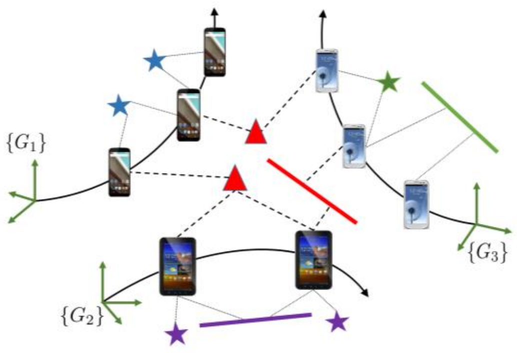
$$\mathcal{C}_{map} = ||\mathbf{z}_k^j - \mathbf{h}(\mathbf{x}_k, \mathbf{p}_{f_j})||^2_{\sigma^2\mathbf{I}} \simeq ||\tilde{\mathbf{z}}_k^j - \mathbf{H}_k^j\tilde{\mathbf{x}}_k - \cancel{\mathbf{H}_f^j\tilde{\mathbf{p}}_{f_j}}||^2_{4\sigma^2\mathbf{I}}$$



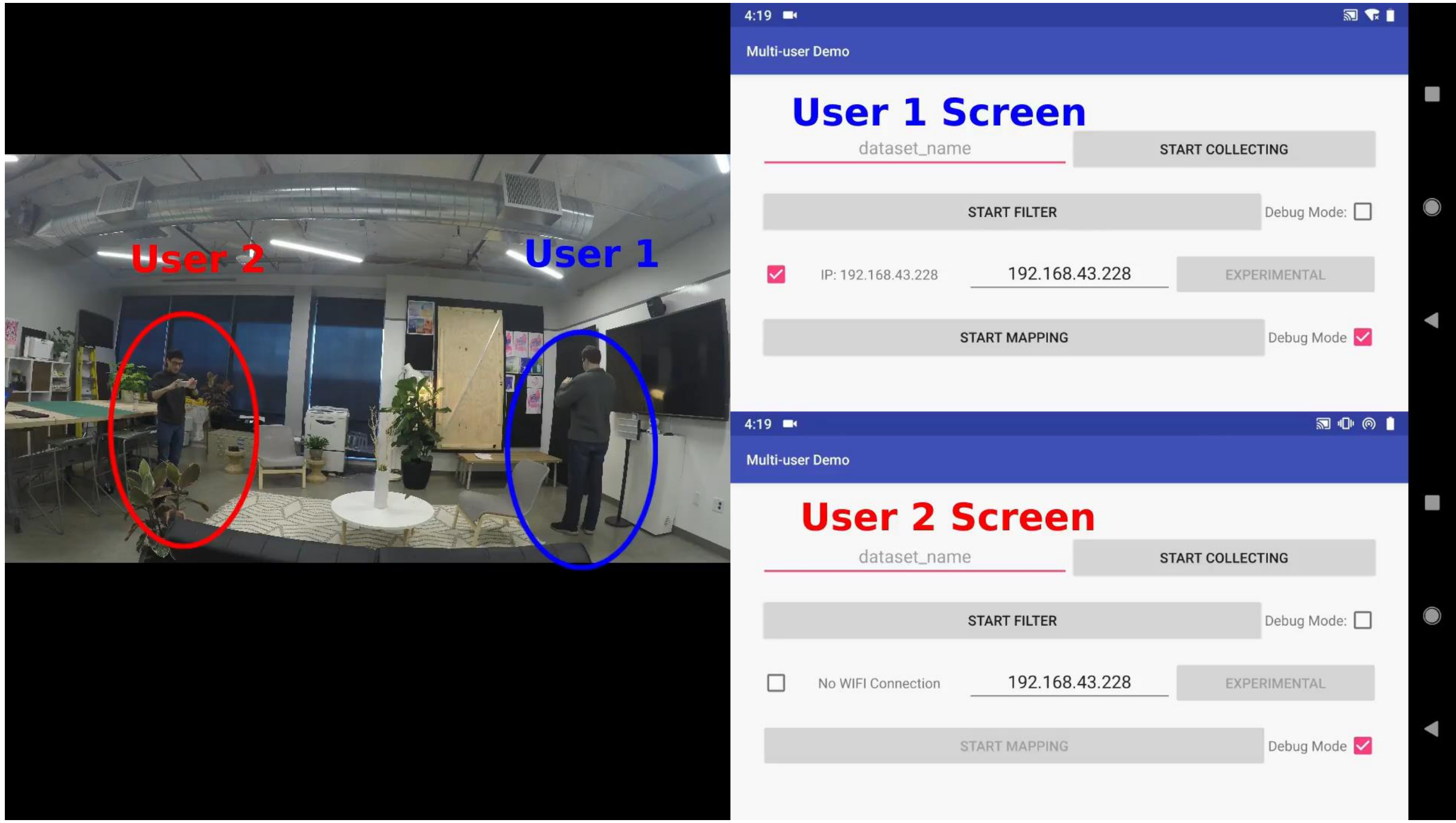
- Consistent alternatives:
  - Schmidt Kalman Filter [3]
  - RISE-SLAM [4]

[1] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "**Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing**," TRO'09
[2] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "**Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization**," RSS'14
[3] R. Dutoit, J. Hesch, E. Nerurkar, and S. Roumeliotis, "**Consistent Map-based 3D Localization on Mobile Devices**," ICRA'17
[4] T. Ke, K. Wu and S. Roumeliotis, "**RISE-SLAM: A Resource-aware Inverse Schmidt Estimator for SLAM**," IROS'19

Cooperative VIO/SLAM

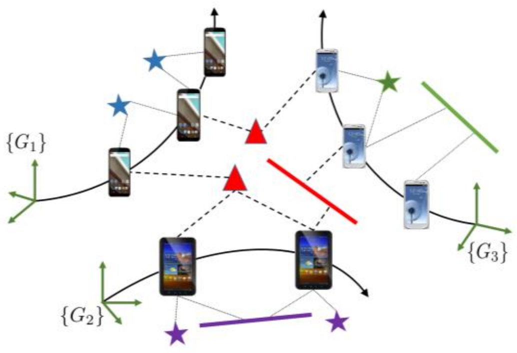


- Data from multiple devices are fused to create an area representation
- Centralized [1,2]
  - Computation is offloaded from device
  - Require powerful server for processing

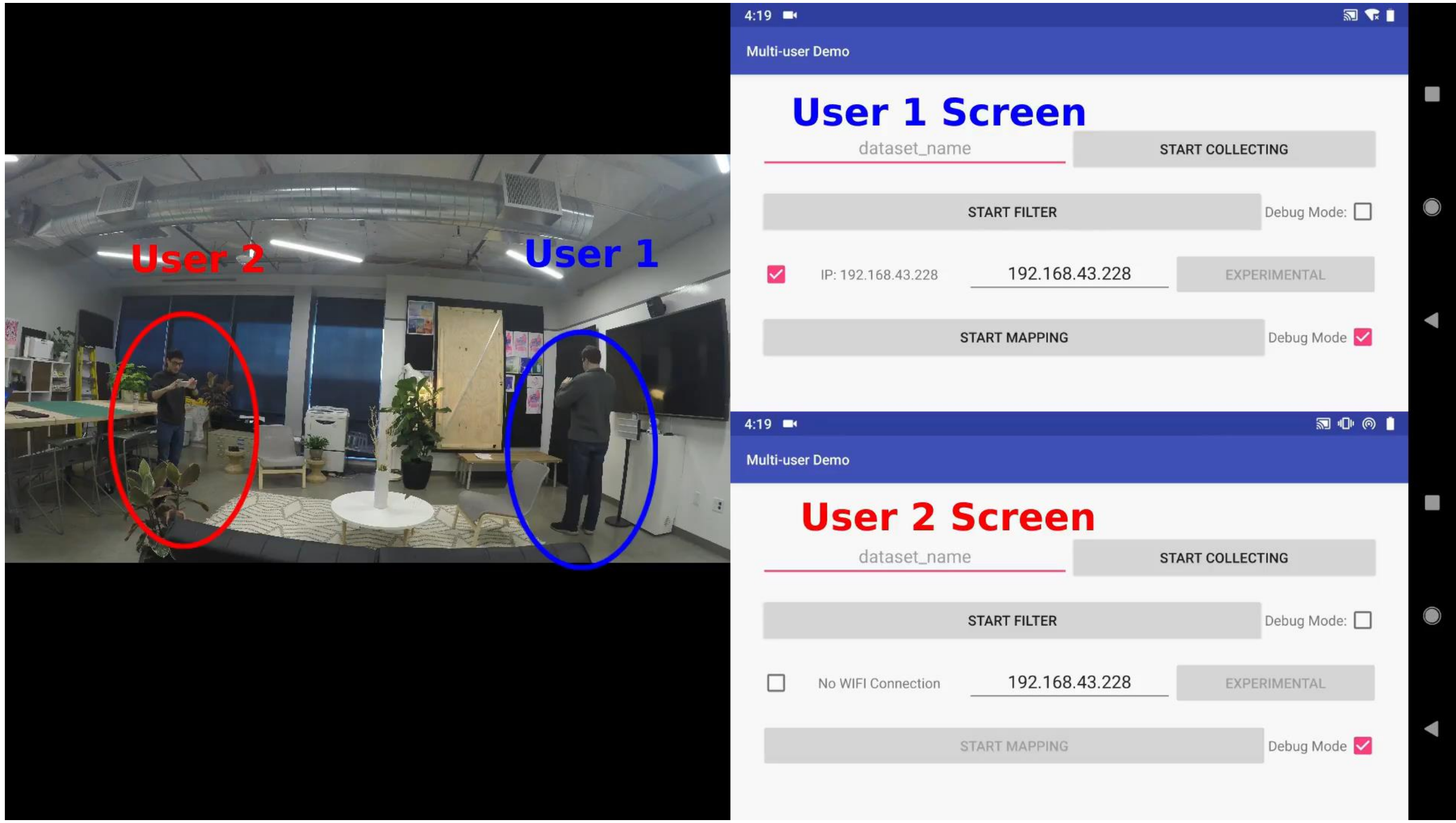


[1] M. Karrer, P. Schmuck, and M. Chli, "**CVI-SLAM collaborative visual-inertial SLAM**," RAL'18
[2] C. Guo, K. Sartipi, R. DuToit, G. Georgiou, R. Li, J. O'Leary, E. Nerurkar, J. Hesch, and S. Roumeliotis, "**Resource-Aware Large-Scale Cooperative Three-Dimensional Mapping Using Multiple Mobile Devices**," TRO'18

Cooperative VIO/SLAM

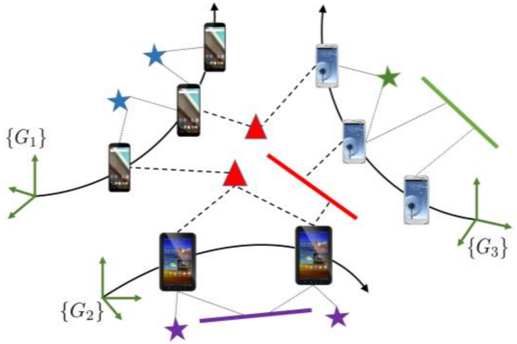


- Data from multiple devices are fused to create an area representation
- Centralized [1,2]
  - Computation is offloaded from device
  - Require powerful server for processing
- Distributed [3,4]
  - All devices cooperate to compute a single area representation

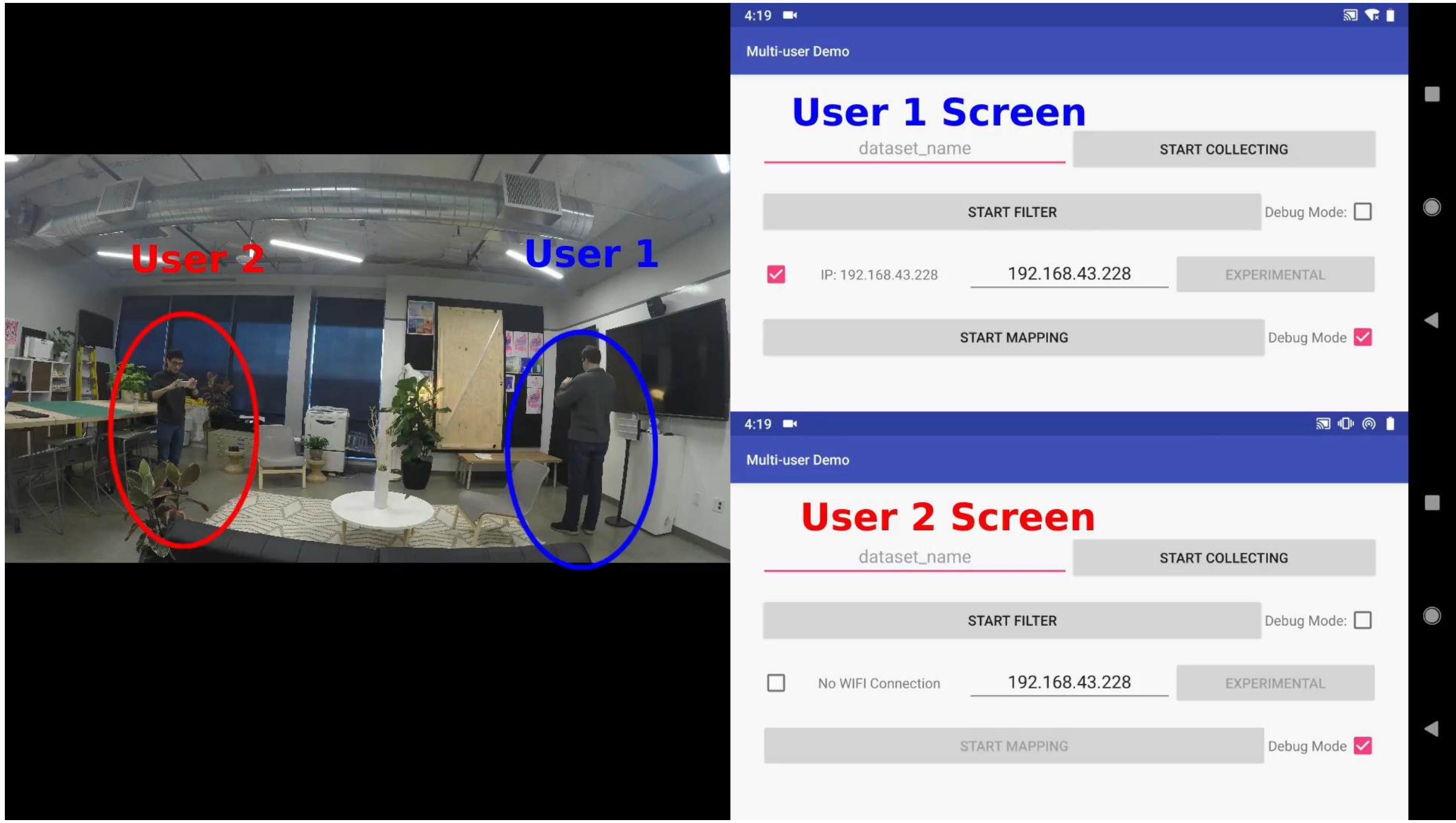


[3] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "**Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models**," IJRR'17
[4] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "**Data-efficient decentralized visual SLAM**," ICRA'18

Cooperative VIO/SLAM



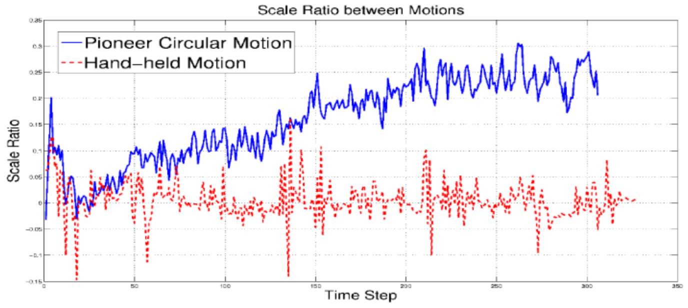
- Data from multiple devices are fused to create an area representation
- Centralized [1,2]
  - Computation is offloaded from device
  - Require powerful server for processing
- Distributed [3,4]
  - All devices cooperate to compute a single area representation
- Multi-centralized [5,6,7]
  - Each device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

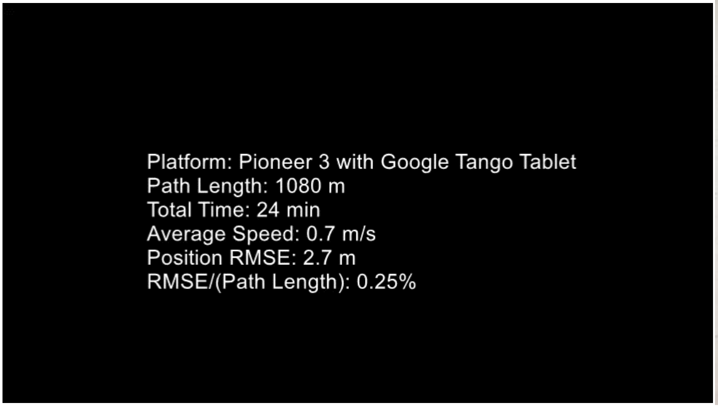


[5] A. Cunningham, V. Indelman, and F. Dellaert, "**DDF-SAM 2.0: Consistent distributed smoothing and mapping**," ICRA'13
[6] H. Zhang, X. Chen, H. Lu, and J. Xiao, "**Distributed and Collaborative Monocular Simultaneous Localization and Mapping for Multi-robot Systems in Large-scale Environments**," IJARS'18
[7] K. Sartipi, R. DuToit, C. Cobar, and S. Roumeliotis, "**Decentralized Visual-Inertial Localization and Mapping on Mobile Devices for Augmented Reality,**" IROS'19

Interesting Research Directions

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

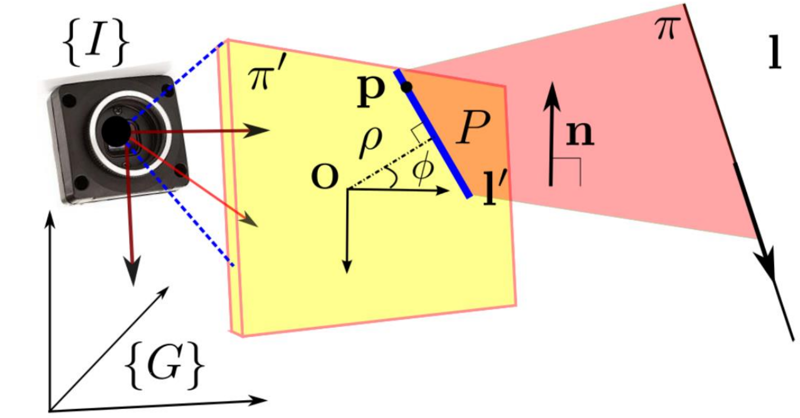




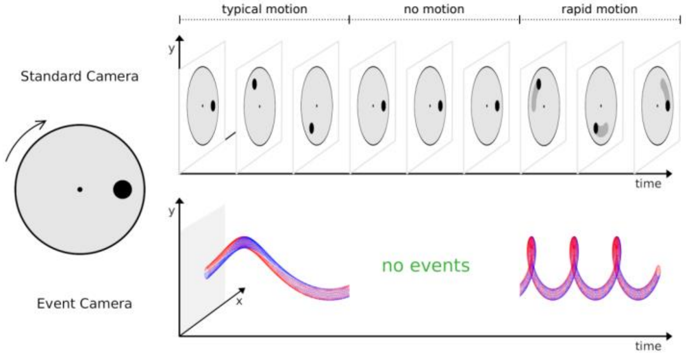
[1] K. J. Wu, C. X. Guo, G. A. Georgiou and S. I. Roumeliotis, “**VINS on Wheels**”, ICRA’17

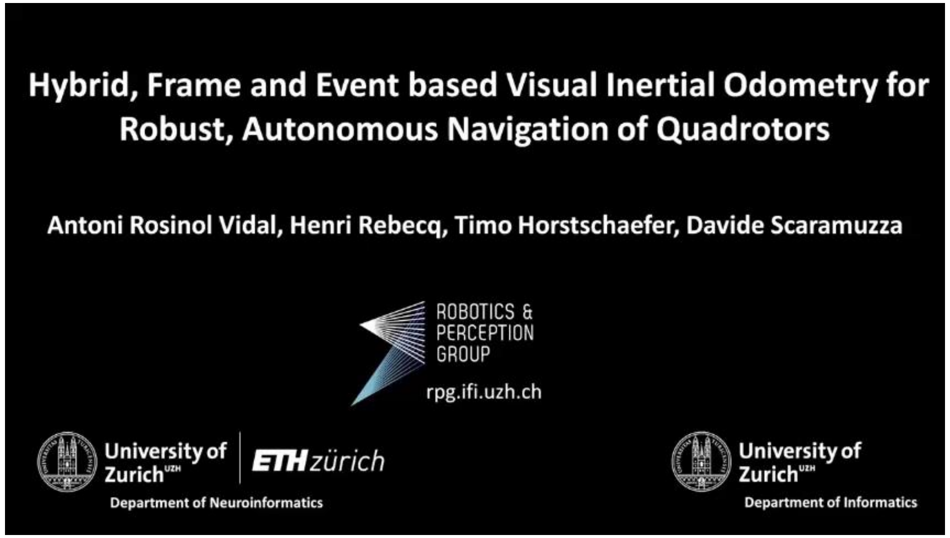
Interesting Research Directions

- **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

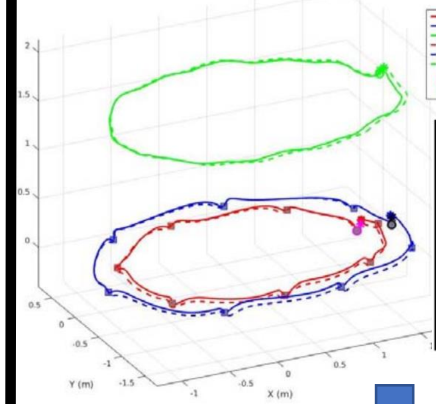


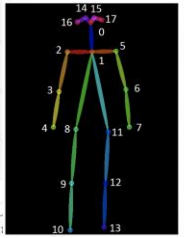
- **Event-based camera** [5,6]
  - Detect changes in intensity, low latency





- **Incorporate system's dynamics**
  - Human motion model [7]





[1] K. J. Wu, C. X. Guo, G. A. Georgiou and S. I. Roumeliotis, "**VINS on Wheels**", ICRA'17
[2] D. G. Kottas and Stergios I. Roumeliotis, "**Exploiting Urban Scenes for Vision-aided Inertial Navigation,**" RSS'13
[3] H. Yu and A. I. Mourikis, "**Vision-Aided Inertial Navigation with Line Features and a Rolling-Shutter Camera**", IROS'15
[4] Y. Yang and G. P. Huang, "**Aided inertial navigation with geometric features: Observability analysis**", ICRA'18
[5] E. Mueggler, G. Gallego, H. Rebecq, D. Scaramuzza, **"Continuous-Time Visual-Inertial Odometry for Event Cameras**," TRO'18
[6] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, **"Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios,"** RA-L'18
[7] A. Ahmed and S. I. Roumeliotis, "**A Visual-Inertial Approach to Human Gait Estimation**", ICRA'18

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

[1] L. Carlone and S. Karaman, "**Attention and anticipation in fast visual-inertial navigation**," TRO'18

[2] D. G. Kottas, R. C. DuToit, A. Ahmed, C. X. Guo, G. A. Georgiou, R. Li and S. I. Roumeliotis, "**A resource-aware vision-aided inertial navigation system for wearable and portable computers**," ICRA'14

[3] K. J. Wu, T. Do, L. C. Carrillo-Arce, and S. I. Roumeliotis, "**On the VINS Resource-Allocation Problem for a Dual-Camera, Small-Size Quadrotor**," ISER'16

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

moving objects concern filtering

movable objects concern mapping

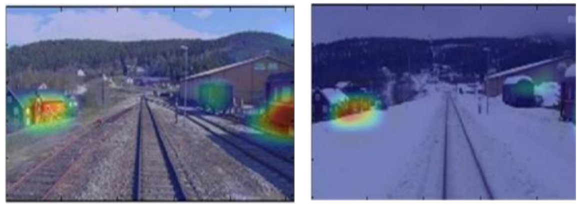
[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "**Mask RCNN**," ICCV'17

[5] T. Pham, T. T. Do, N. Sünderhauf, and I. Reid, "**Scenecut: Joint geometric and object segmentation for indoor scenes**," ICRA'18

[6] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid and M. Milford, "**Deep Learning Features at Scale for Visual Place Recognition**," ICRA'17

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

Season/light Invariance



Viewpoint Invariance

Thank You!

MULTIPLE AUTONOMOUS ROBOTIC SYSTEMS LABORATORY