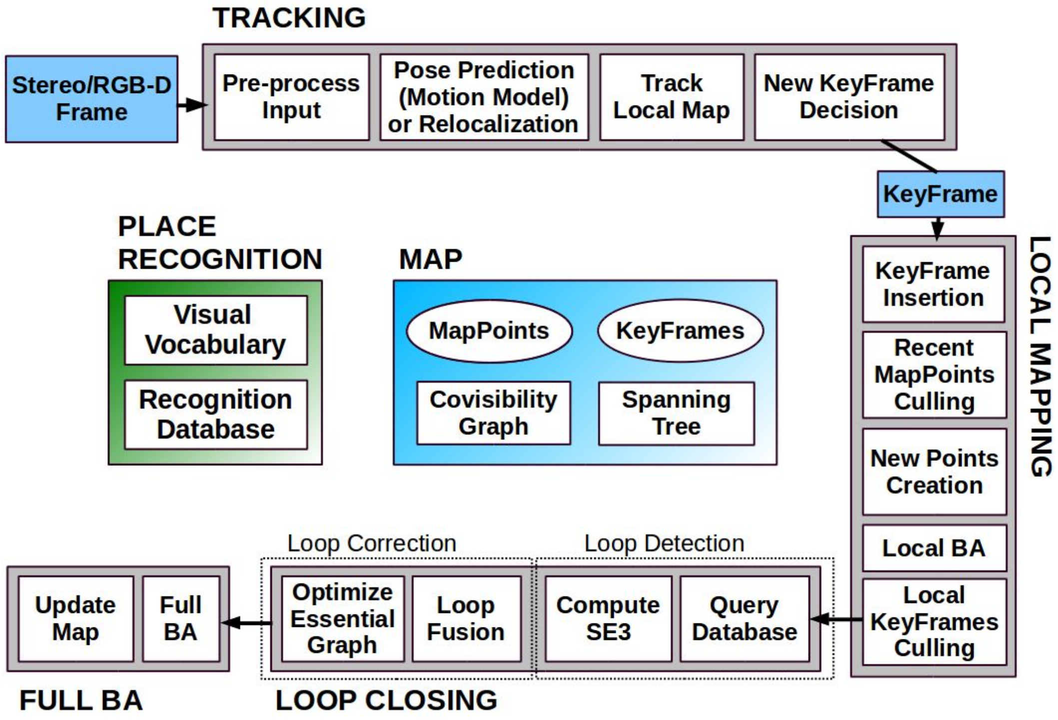
ORB-SLAM2 with IMU Preintegration

Group 10
Joseph Yates, Deyang Dai, Yizhou Lu, Xiaotong Chen

UNIVERSITY OF MICHIGAN

ORB-SLAM2

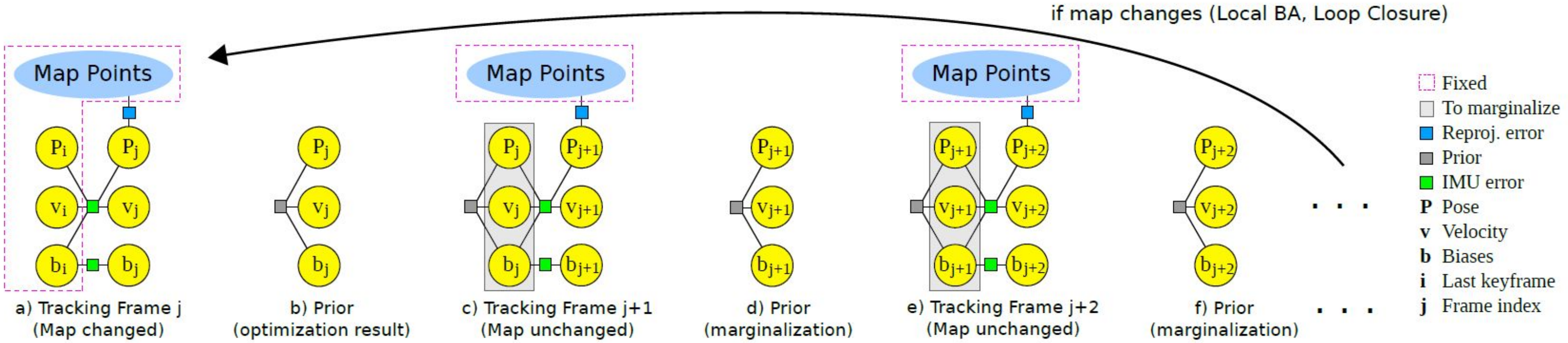


Four Threads

1. Tracking (main thread)
2. Local Mapping
3. Loop Closing
4. Viewer

Ref: Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics,* vol. 33, no. 5, pp. 1255-1262, 2017.

Tracking



Map Changed:

$$\theta = \left\{ \mathbf{R}_{\mathtt{WB}}^{j}, {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^{j}, {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^{j}, \mathbf{b}_{g}^{j}, \mathbf{b}_{a}^{j} \right\}$$

$$\theta^{*} = \underset{\theta}{\operatorname{argmin}} \left(\sum_{k} \mathbf{E}_{\mathrm{proj}}(k,j) + \mathbf{E}_{\mathrm{IMU}}(i,j) \right)$$

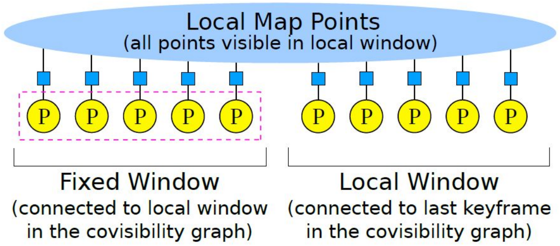
Map Unchanged:

$$\theta = \left\{ \mathbf{R}_{\mathtt{WB}}^{j}, \mathbf{p}_{\mathtt{W}}^{j}, \mathbf{v}_{\mathtt{W}}^{j}, \mathbf{b}_{g}^{j}, \mathbf{b}_{a}^{j}, \mathbf{R}_{\mathtt{WB}}^{j+1}, \mathbf{p}_{\mathtt{W}}^{j+1}, \mathbf{v}_{\mathtt{W}}^{j+1}, \mathbf{b}_{g}^{j+1}, \mathbf{b}_{a}^{j+1} \right\}$$

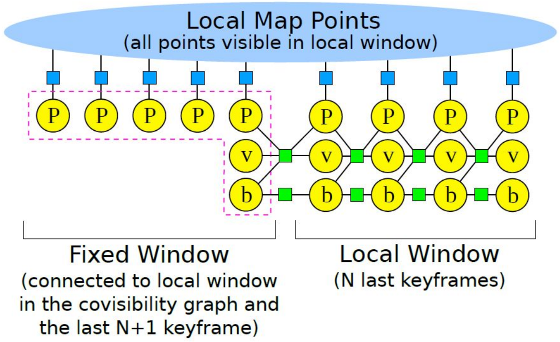
$$\theta^{*} = \underset{\theta}{\operatorname{argmin}} \left(\sum_{k} \mathbf{E}_{\mathrm{proj}}(k,j+1) + \mathbf{E}_{\mathrm{IMU}}(j,j+1) + \mathbf{E}_{\mathrm{prior}}(j) \right)$$

Ref: Mur-Artal, Raúl, and Juan D. Tardós. "Visual-inertial monocular SLAM with map reuse." IEEE Robotics and Automation Letters 2.2 (2017): 796-803.

Local Mapping



ORB-SLAM's Local BA



Visual-Inertial ORB-SLAM's Local BA

- Performs local BA after a new keyframe insertion
- Optimizes the last N keyframes (local window) and map points
- A suitable local window size to reduce complexity

Ref: Mur-Artal, Raúl, and Juan D. Tardós. "Visual-inertial monocular SLAM with map reuse." IEEE Robotics and Automation Letters 2.2 (2017): 796-803.

IMU Model

- We have a system state with *velocity* and *bias*:

$$\mathbf{x}_i \doteq [\mathrm{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i]$$

- We have a model for the noise and bias the IMU adds to the true *linear acceleration* and *angular velocity* of the system:

$$\begin{aligned} {}_{\mathrm{B}}\tilde{\boldsymbol{\omega}}_{\mathrm{WB}}(t) &= {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}}(t) + \mathbf{b}^g(t) + \boldsymbol{\eta}^g(t) \\ {}_{\mathrm{B}}\tilde{\mathbf{a}}(t) &= \mathrm{R}_{\mathrm{WB}}^{\mathsf{T}}(t)\left({}_{\mathrm{W}}\mathbf{a}(t) - {}_{\mathrm{W}}\mathbf{g}\right) + \mathbf{b}^a(t) + \boldsymbol{\eta}^a(t) \end{aligned}$$

- We can simply propagate forward in (discrete) time using a method not-unlike the Bayes Filter/Kalman Filter prediction step:

$$\begin{aligned} \mathrm{R}(t+\Delta t) &= \mathrm{R}(t)\ \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}(t) - \mathbf{b}^g(t) - \boldsymbol{\eta}^{gd}(t)\right)\Delta t\right) \\ \mathbf{v}(t+\Delta t) &= \mathbf{v}(t) + \mathbf{g}\Delta t + \mathrm{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t \\ \mathbf{p}(t+\Delta t) &= \mathbf{p}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 \\ &\quad + \frac{1}{2}\mathrm{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t^2, \end{aligned}$$

Different Rates

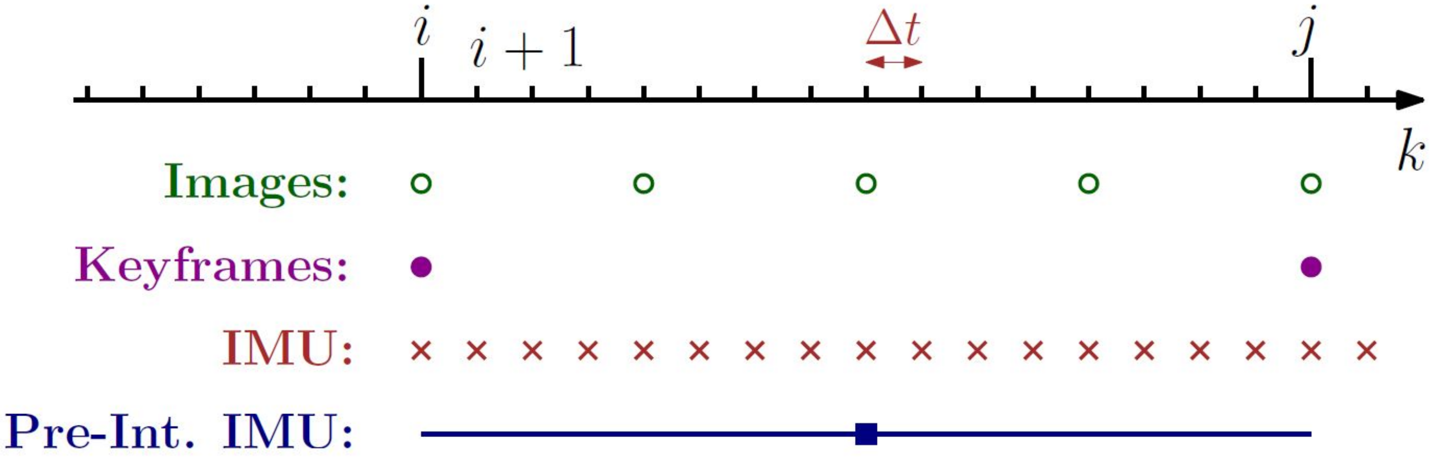


Fig. Different rates for IMU and camera

IMU Preintegration on Manifold

- We want to summarize the IMU measurements at all timesteps k between keyframes at times i and j, independent of the state estimates at i and j - the solver will change states i and j every keyframe
- Seek quantities that summarizes the *relative motion* between keyframes

Idealized Motion Model

$$\Delta \mathrm{R}_{ij} \doteq \mathrm{R}_i^\mathsf{T} \mathrm{R}_j = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right) \Delta t\right)$$

$$\Delta \mathbf{v}_{ij} \doteq \mathrm{R}_i^\mathsf{T} \left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) = \sum_{k=i}^{j-1} \Delta \mathrm{R}_{ik} \left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right) \Delta t$$

$$\Delta \mathbf{p}_{ij} \doteq \mathrm{R}_i^\mathsf{T} \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \tfrac{1}{2} \textstyle\sum_{k=i}^{j-1} \mathbf{g}\Delta t^2\right)$$

$$= \sum_{k=i}^{j-1} \left[\Delta \mathbf{v}_{ik} \Delta t + \frac{1}{2} \Delta \mathrm{R}_{ik} \left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right) \Delta t^2\right]$$

Measurement Model

$$\Delta \tilde{\mathrm{R}}_{ij} = \mathrm{R}_i^\mathsf{T} \mathrm{R}_j \mathrm{Exp}\left(\delta \boldsymbol{\phi}_{ij}\right)$$

$$\Delta \tilde{\mathbf{v}}_{ij} = \mathrm{R}_i^\mathsf{T} \left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) + \delta \mathbf{v}_{ij}$$

$$\Delta \tilde{\mathbf{p}}_{ij} = \mathrm{R}_i^\mathsf{T} \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g}\Delta t_{ij}^2\right) + \delta \mathbf{p}_{ij}$$

IMU Factors

- IMU factors are formed with *residuals* calculated between the measured relative motion quantities and calculated relative motion quantities, which are functions of the state estimates (decision variables)
- They also require *Jacobians*, partial derivatives of the residuals with respect to the state estimate noise quantities

$$\mathbf{r}_{\Delta \mathrm{R}_{ij}} \doteq \mathrm{Log}\left(\left(\Delta\tilde{\mathrm{R}}_{ij}(\bar{\mathbf{b}}_i^g)\mathrm{Exp}\left(\frac{\partial \Delta\bar{\mathrm{R}}_{ij}}{\partial \mathbf{b}^g}\delta\mathbf{b}^g\right)\right)^{\mathsf{T}} \mathrm{R}_i^{\mathsf{T}}\mathrm{R}_j\right)$$

$$\mathbf{r}_{\Delta \mathbf{v}_{ij}} \doteq \mathrm{R}_i^{\mathsf{T}}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - \left[\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^g}\delta\mathbf{b}^g + \frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}^a}\delta\mathbf{b}^a\right]$$

$$\mathbf{r}_{\Delta \mathbf{p}_{ij}} \doteq \mathrm{R}_i^{\mathsf{T}}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - \left[\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}^g}\delta\mathbf{b}^g + \frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}_a}\delta\mathbf{b}^a\right]$$

$$\frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\phi_i} = (\mathrm{R}_i^{\mathsf{T}}(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2))^\wedge \qquad \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\phi_j} = 0$$

$$\frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\mathbf{p}_i} = -\mathbf{I}_{3\times 1} \qquad \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\mathbf{p}_j} = \mathrm{R}_i^{\mathsf{T}}\mathrm{R}_j$$

$$\frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\mathbf{v}_i} = -\mathrm{R}_i^{\mathsf{T}}\Delta t_{ij} \qquad \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\mathbf{v}_j} = 0$$

$$\frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\mathbf{b}_i^a} = -\frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}_i^a} \qquad \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \delta\mathbf{b}_i^g} = -\frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}_i^g}$$

Plus Jacobians for velocity and orientation (similar in format to the above)

Noise Propagation

IMU Integration

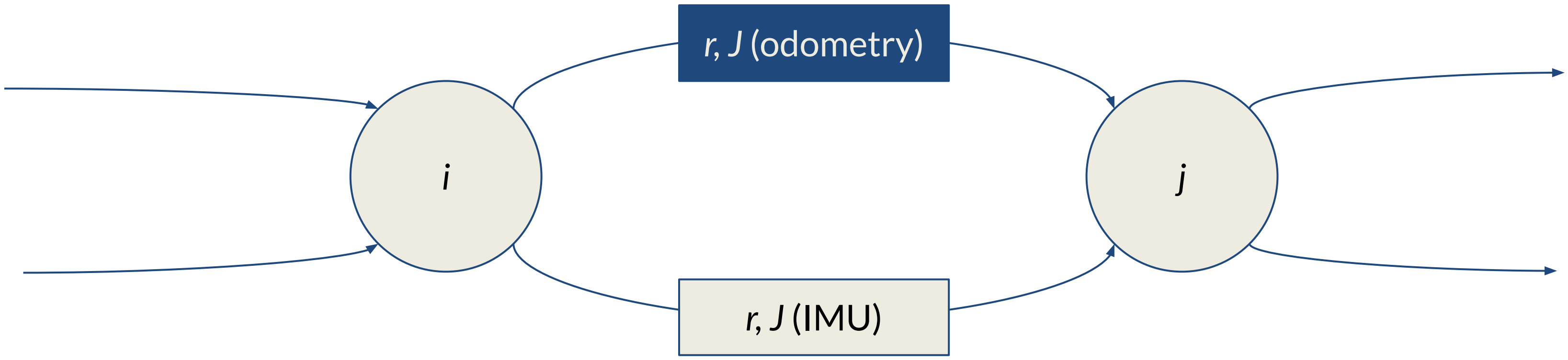
$$X_{k+1} = f(X_k, a_k, \omega_k)$$

Covariance update

$$\Sigma_{k+1} = A_k \Sigma_k A_k^T + B_k \Sigma_\eta^a B_k^T + C_k \Sigma_\eta^g C_k^T$$

Factor Graph

- We may then build a factor graph using (just!) IMU data.
- We may also take another solution, which did not use its IMU data, convert its solved poses into odometry factors, and demonstrate that the IMU factors improve the overall solution relative to the ground truth



Gauss-Newton Solver

General Algorithm:

1. Given a state vector containing two keyframes $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$

2. Calculate residual and Jacobian
$$r = f_1(X_1, X_2, IMU\ measurment)$$
$$J = f_2(X_1, X_2)$$

3. Solve for the optimized step
$$\delta X = J^T (J * J^T)^{-1} r$$

4. Update state vector
$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}_{k+1} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}_k + \delta X$$

5. Exit the loop after step convergence
$$\delta X < \epsilon$$

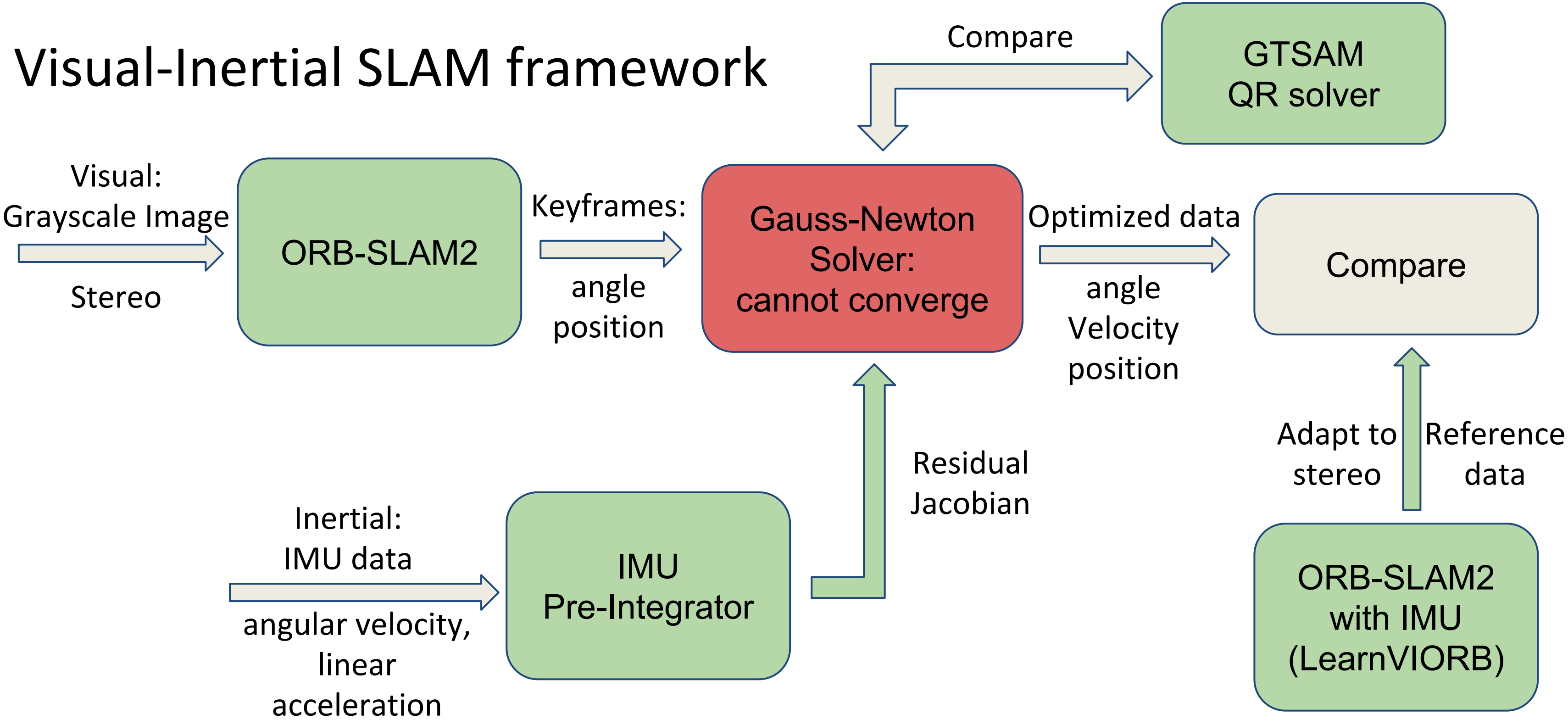
Gauss-Newton Solver

Jacobian matrix

$$\begin{bmatrix} \frac{\partial r_{R12}}{\partial \delta X_1} & \frac{\partial r_{R12}}{\partial \delta X_2} & 0 & 0 \\ \frac{\partial r_{v12}}{\partial \delta X_1} & \frac{\partial r_{v12}}{\partial \delta X_2} & 0 & 0 \\ \frac{\partial r_{p12}}{\partial \delta X_1} & \frac{\partial r_{p12}}{\partial \delta X_2} & 0 & 0 \\ 0 & \frac{\partial r_{R23}}{\partial \delta X_2} & \frac{\partial r_{R23}}{\partial \delta X_3} & 0 \\ 0 & \frac{\partial r_{v23}}{\partial \delta X_2} & \frac{\partial r_{v23}}{\partial \delta X_3} & 0 \\ 0 & \frac{\partial r_{p23}}{\partial \delta X_2} & \frac{\partial r_{p23}}{\partial \delta X_3} & 0 \\ 0 & 0 & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \delta X_1 \\ \delta X_2 \\ \delta X_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} r_{R12} \\ r_{v12} \\ r_{p12} \\ r_{R23} \\ r_{v23} \\ r_{p23} \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

Current Progress

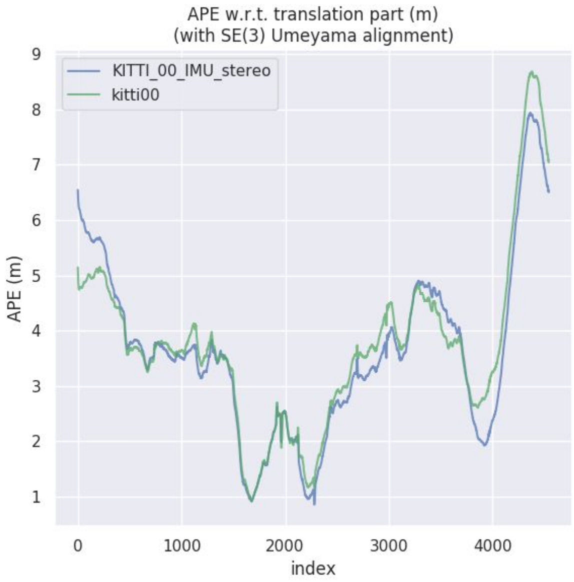
Visual-Inertial SLAM framework

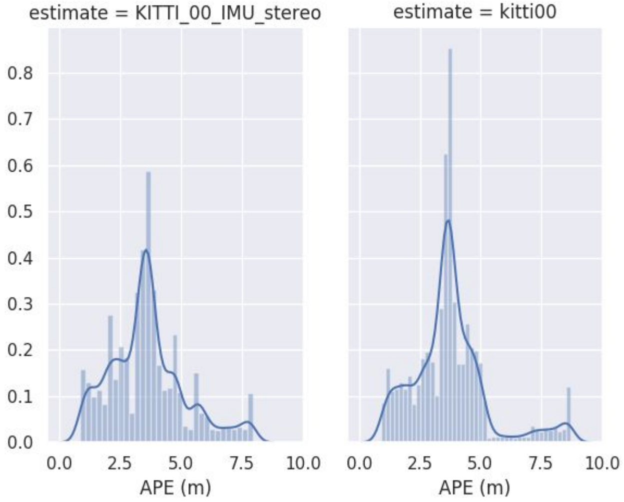


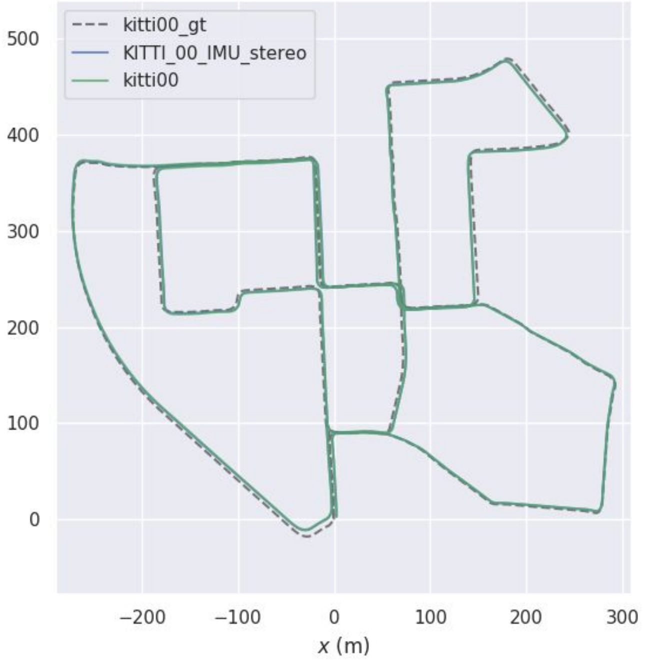
Result

LearnVIORB performance improvement

- Evo: evaluation with Umeyama SE(3) alignment (~3% improve)





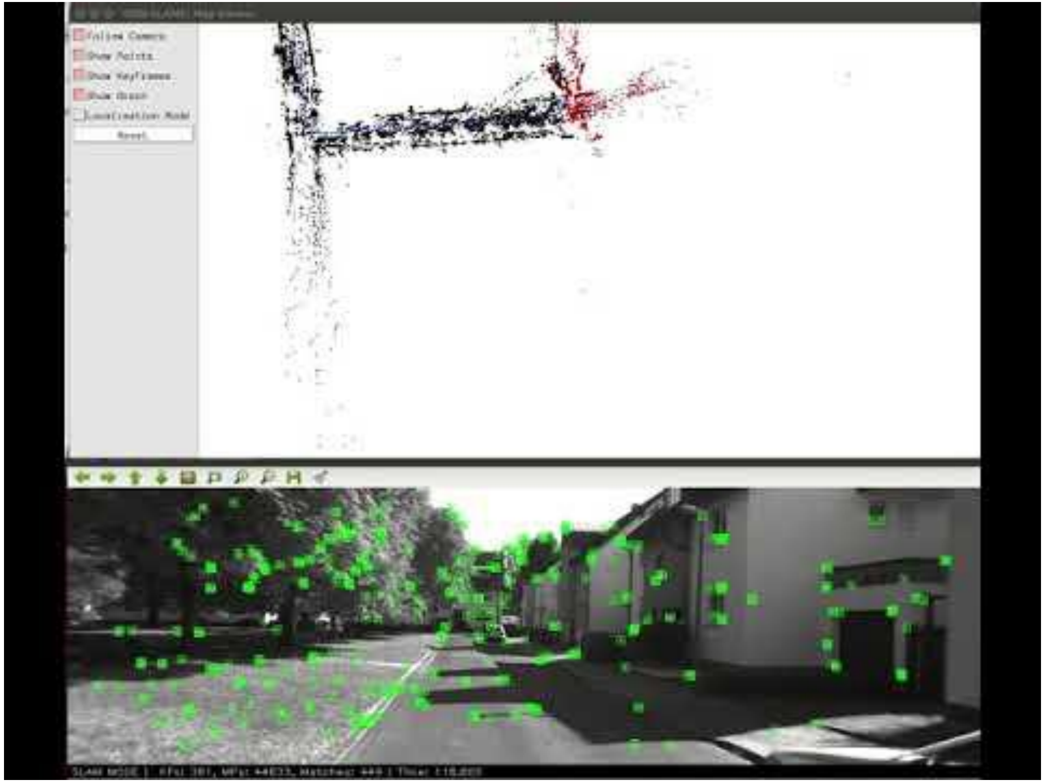


https://github.com/jingpang/LearnVIORB; https://github.com/MichaelGrupp/evo

Result

LearnVIORB performance

- Stereo
- 10Hz IMU
- 10Hz Image

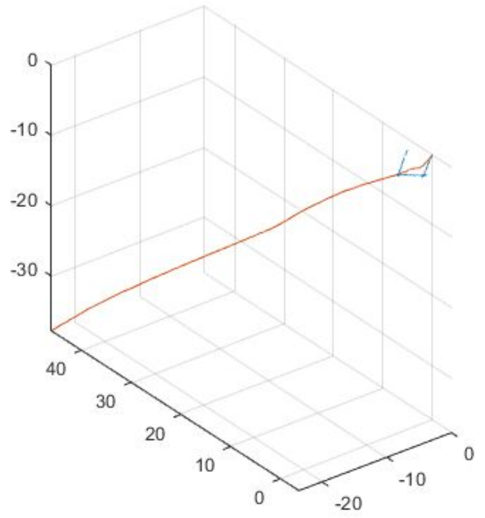


https://github.com/jingpang/LearnVIORB; https://github.com/MichaelGrupp/evo

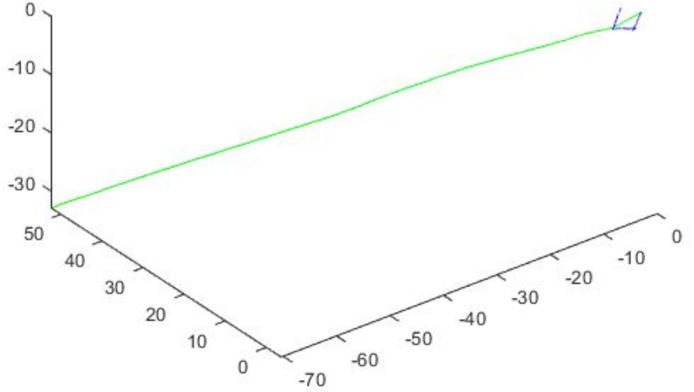
Result

- Initial results for accuracy of IMU factors still pending GN/QR solver
- Baselines available from simple propagation using IMU model and using an IMU factor-only SAM solution from GTSAM (QR factorization)

IMU Propagation Model vs. Ground Truth



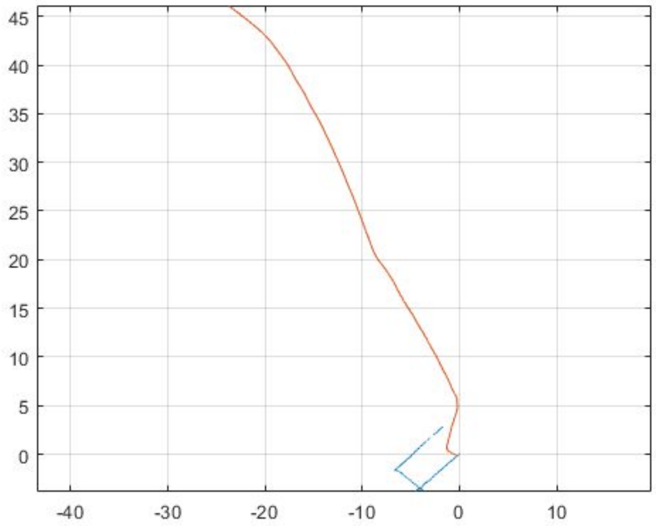
GTSAM Factor Graph Solution vs. Ground Truth



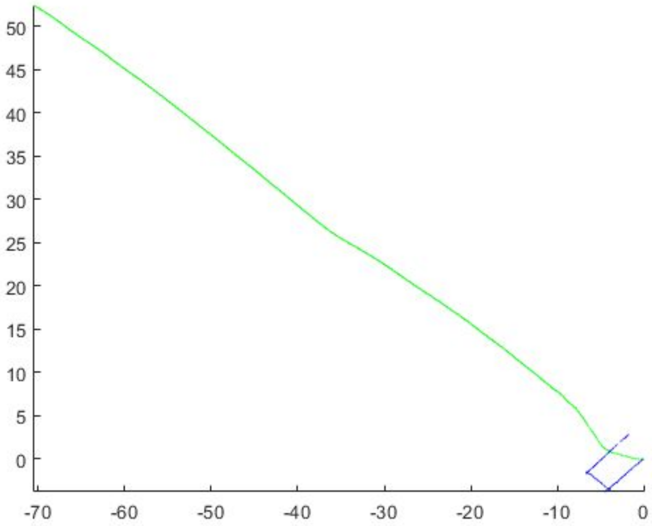
Result

- Initial results for accuracy of IMU factors still pending GN/QR solver
- Baselines available from simple propagation using IMU model and using an IMU factor-only SAM solution from GTSAM (QR factorization)

IMU Propagation Model vs. Ground Truth



GTSAM Factor Graph Solution vs. Ground Truth



Ongoing Work

- Debugging Gauss-Newton solver - switching to QR factorization may help
- Adding odometry factors to our implementation and the GTSAM baseline for comparison
- Testing our implementation vs. GTSAM on KAIST Urban Dataset
- Compare final results to ORB-SLAM2 with LearnVIORB on KAIST and evaluate performance