Exchange & Share about VINS

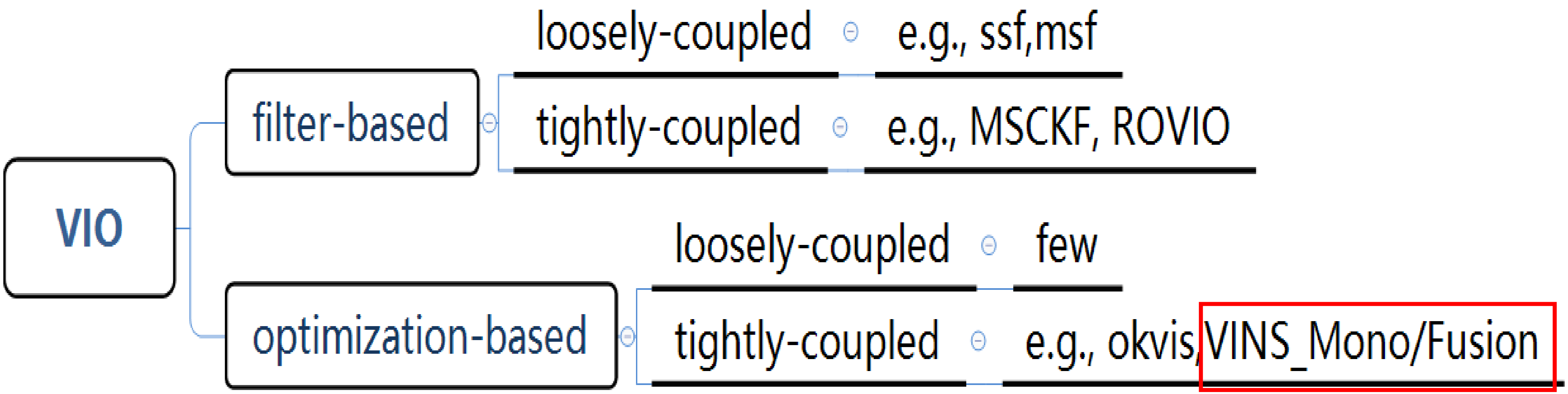
Tao Zhang

2019.6.6

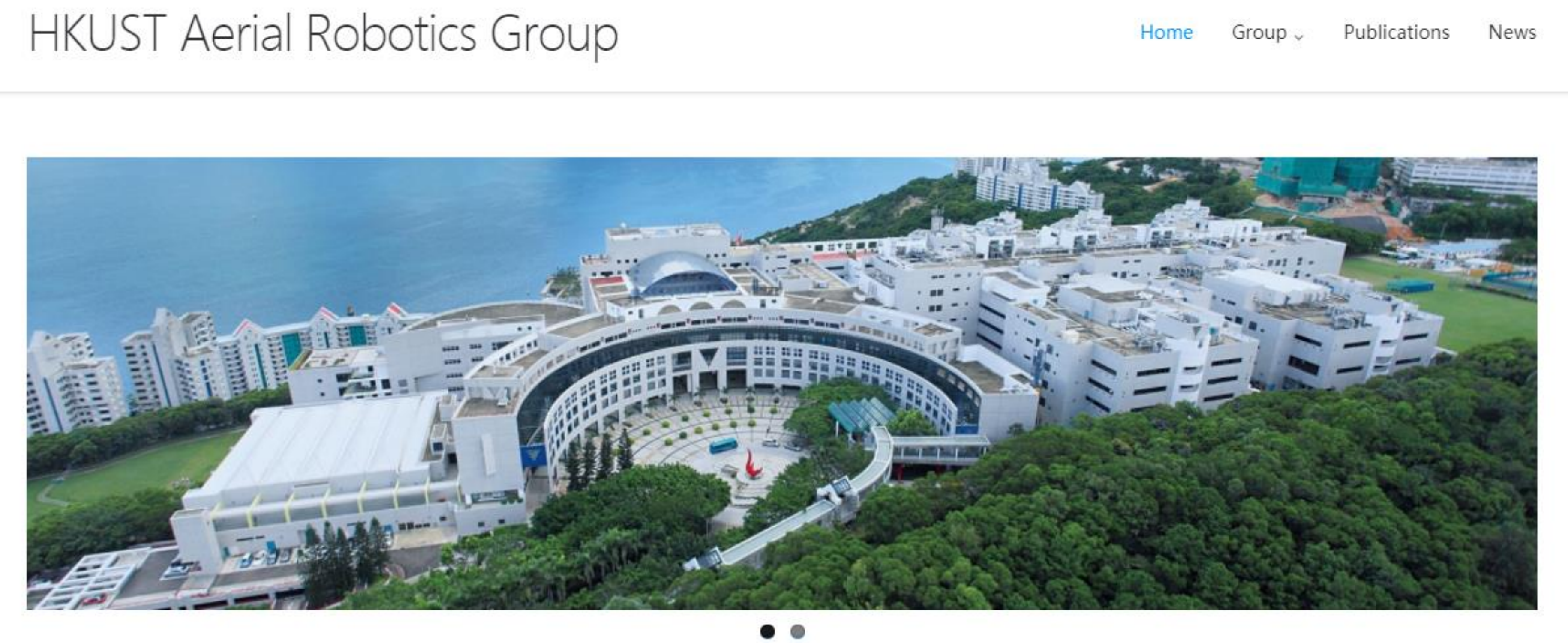
◆VIO

- **VIO**: Vusual Inertial Odometer

- Classification

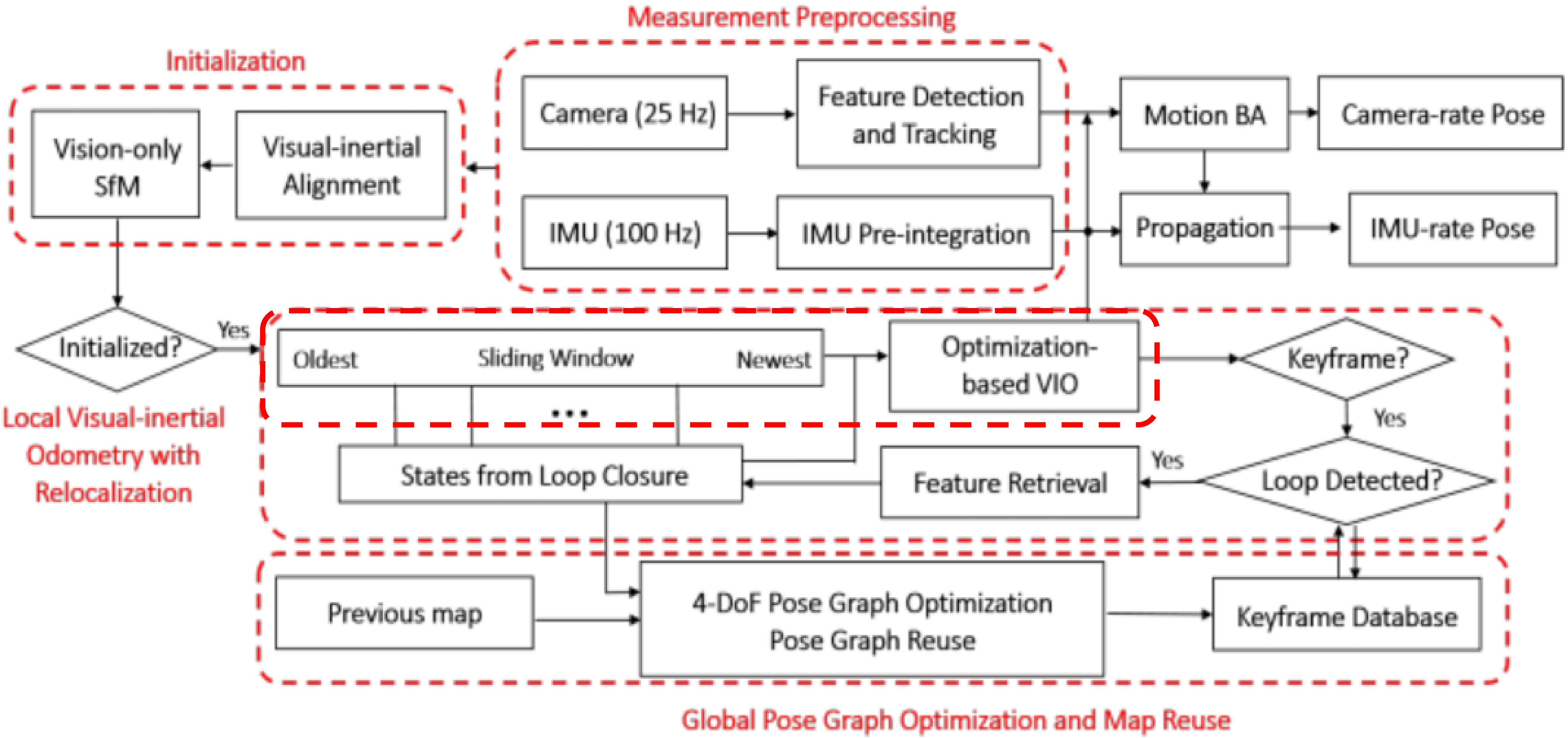


◆Background

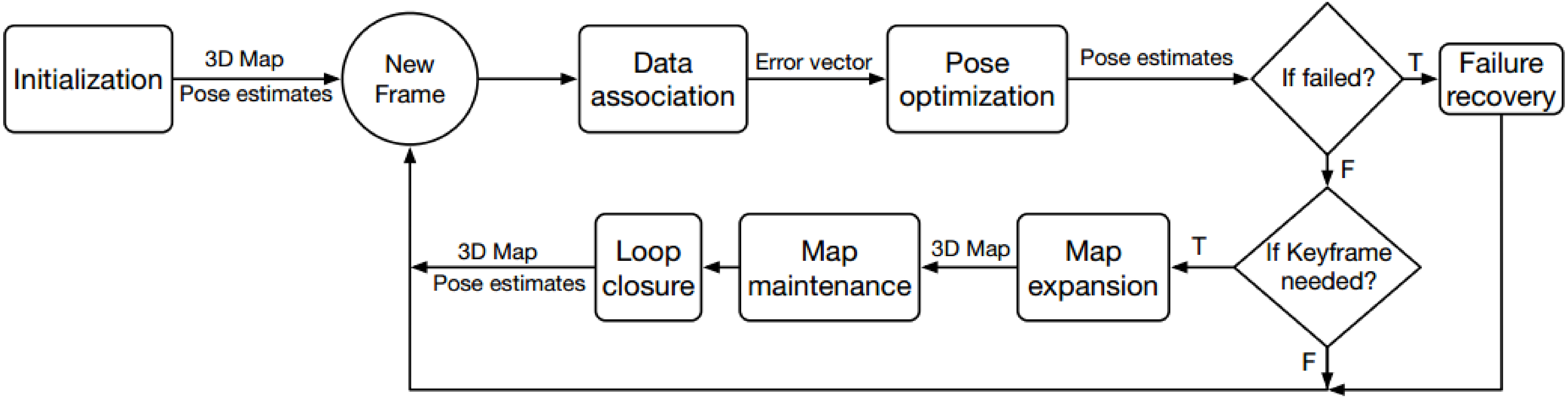


- VINS_Mono： VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. 2017.8
- VINS_Fusion： A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors
  A General Optimization-based Framework for Global Pose Estimation with Multiple Sensors 2019.1

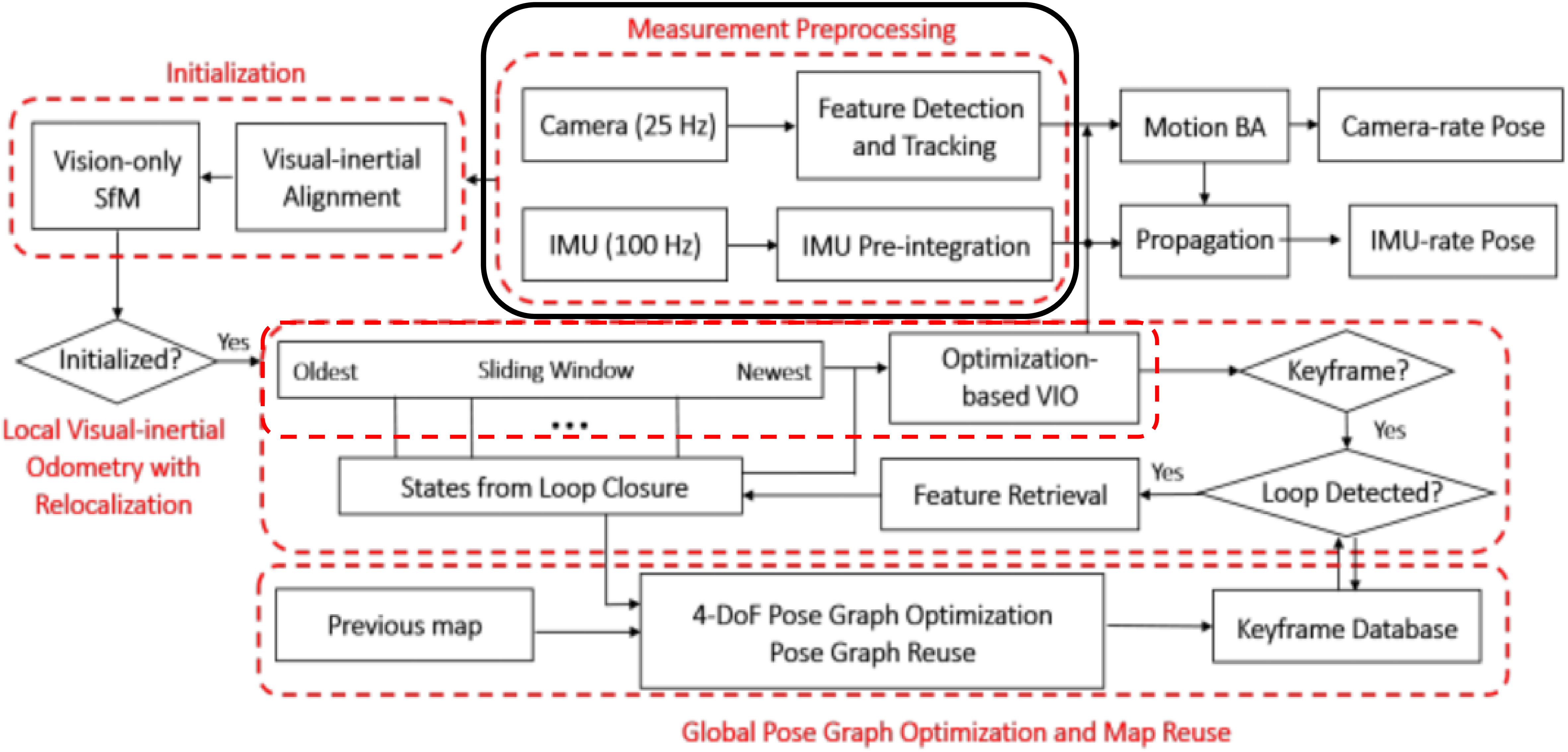
◆VINS_Mono Frame



◆Keyframe-SLAM flowchart



◆Measurement Preprocessing



◆vison front-end

- Feature processing

-Harris corners (goodFeaturesToTrack() in OpenCV)

-KLT sparse optical flow tracker(calcOpticalFlowPyrLK() in OpenCV)

-RANSAC outlier rejection

- Keyframe selection

-average parallax method(平均视差法)

Rotation-compensated average feature parallax is larger than a threshold

-tracking quality method(跟踪质量法)

Number of tracked features in the current frame is less than a threshold

◆IMU Pre-integration

$$\hat{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{b}_{a_t} + \mathbf{R}^t_w \mathbf{g}^w + \mathbf{n}_a$$
$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t + \mathbf{b}_{w_t} + \mathbf{n}_w. \quad (1)$$

raw measurements

$$\begin{aligned}
\mathbf{p}^w_{b_{k+1}} &= \mathbf{p}^w_{b_k} + \mathbf{v}^w_{b_k}\Delta t_k \\
&\quad + \iint_{t\in[t_k,t_{k+1}]} \left(\boxed{\mathbf{R}^w_t}(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a) - \mathbf{g}^w\right) dt^2 \\
\mathbf{v}^w_{b_{k+1}} &= \mathbf{v}^w_{b_k} + \int_{t\in[t_k,t_{k+1}]} \left(\boxed{\mathbf{R}^w_t}(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a) - \mathbf{g}^w\right) dt \\
\mathbf{q}^w_{b_{k+1}} &= \boxed{\mathbf{q}^w_{b_k}} \otimes \int_{t\in[t_k,t_{k+1}]} \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{w_t} - \mathbf{n}_w)\mathbf{q}^{b_k}_t \, dt,
\end{aligned} \quad (3)$$

integration in word frame**(traditional method)**

◆IMU Pre-integration

$$\mathbf{R}_w^{b_k}\mathbf{p}_{b_{k+1}}^w = \mathbf{R}_w^{b_k}(\mathbf{p}_{b_k}^w + \mathbf{v}_{b_k}^w\Delta t_k - \frac{1}{2}\mathbf{g}^w\Delta t_k^2) + \boldsymbol{\alpha}_{b_{k+1}}^{b_k}$$
$$\mathbf{R}_w^{b_k}\mathbf{v}_{b_{k+1}}^w = \mathbf{R}_w^{b_k}(\mathbf{v}_{b_k}^w - \mathbf{g}^w\Delta t_k) + \boldsymbol{\beta}_{b_{k+1}}^{b_k}$$
$$\mathbf{q}_w^{b_k} \otimes \mathbf{q}_{b_{k+1}}^w = \boldsymbol{\gamma}_{b_{k+1}}^{b_k}, \tag{5}$$

Process the integration from one imu frame to another imu frame

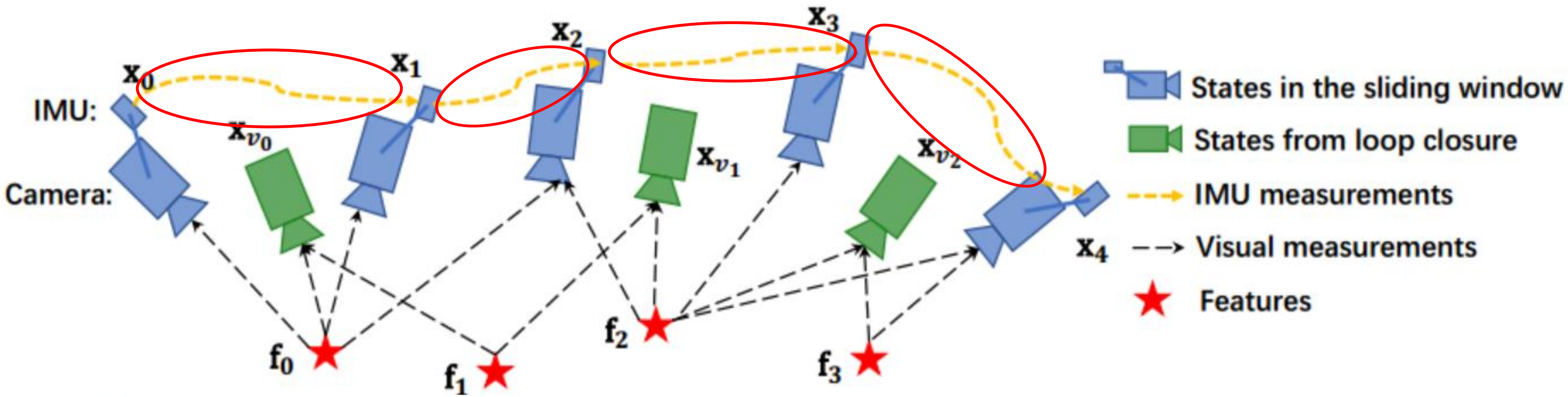
$$\boldsymbol{\alpha}_{b_{k+1}}^{b_k} = \iint_{t\in[t_k,t_{k+1}]} \mathbf{R}_t^{b_k}(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a)dt^2$$
$$\boldsymbol{\beta}_{b_{k+1}}^{b_k} = \int_{t\in[t_k,t_{k+1}]} \mathbf{R}_t^{b_k}(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a)dt \tag{6}$$
$$\boldsymbol{\gamma}_{b_{k+1}}^{b_k} = \int_{t\in[t_k,t_{k+1}]} \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{w_t} - \mathbf{n}_w)\boldsymbol{\gamma}_t^{b_k}dt.$$

Pre-integration(s.t., the imu measurement)

discretization、convariance matrix、jacobian matrix

·········and so much work

◆IMU Pre-integration



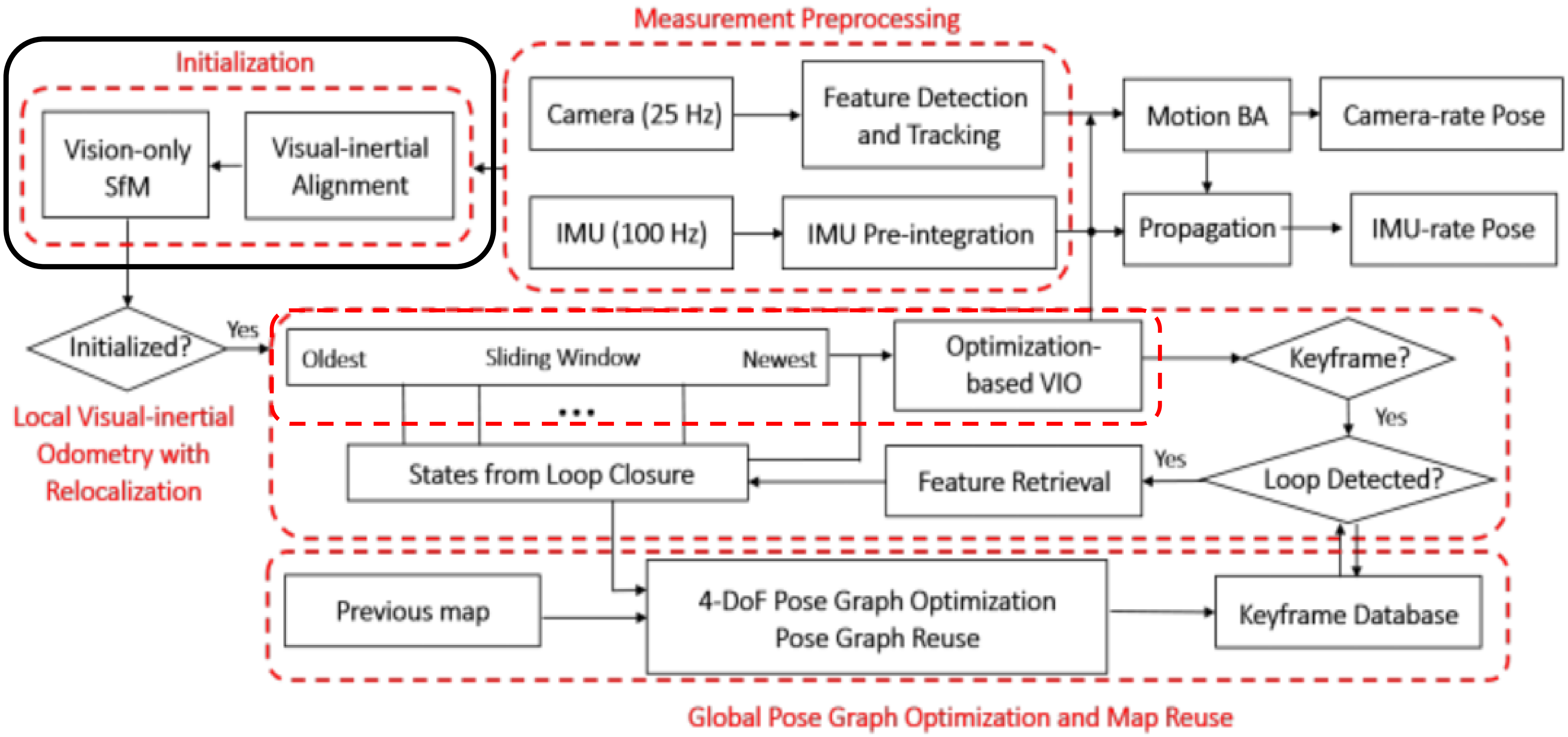
$$\begin{bmatrix} \hat{\alpha}^{b_k}_{b_{k+1}} \\ \hat{\beta}^{b_k}_{b_{k+1}} \\ \hat{\gamma}^{b_k}_{b_{k+1}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{b_k}_w(\mathbf{p}^w_{b_{k+1}} - \mathbf{p}^w_{b_k} + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 - \mathbf{v}^w_{b_k}\Delta t_k) \\ \mathbf{R}^{b_k}_w(\mathbf{v}^w_{b_{k+1}} + \mathbf{g}^w \Delta t_k - \mathbf{v}^w_{b_k}) \\ {\mathbf{q}^w_{b_k}}^{-1} \otimes \mathbf{q}^w_{b_{k+1}} \\ \mathbf{b}_{ab_{k+1}} - \mathbf{b}_{ab_k} \\ \mathbf{b}_{wb_{k+1}} - \mathbf{b}_{wb_k} \end{bmatrix}. \tag{13}$$

the IMU measurement model based on pre-integration

measurement part

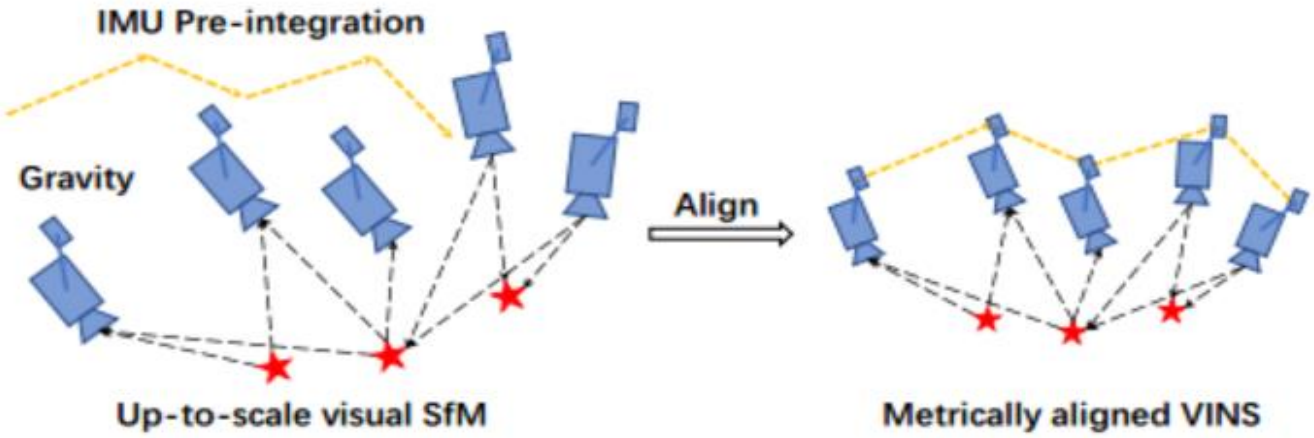
estimated part

◆Initialization



◆Initialization

- Why? Tightly-coupled visual-inertial odometry is a highly nonlinear system;
- Assumption:
  1. known camera intrinsic and camera-IMU extrinsic before initialization
  2. known accelerometer and gyroscope biases initial value
- **Initialization: a loosely-coupled sensor fusion process**(vision-only SLAM + visual-inertial alignment)
- ignore accelerometer bias terms in the initial step



◆vision-only SLAM(SfM)

- Sliding Window Vision-Only SLAM

  -in small window(10 fps)

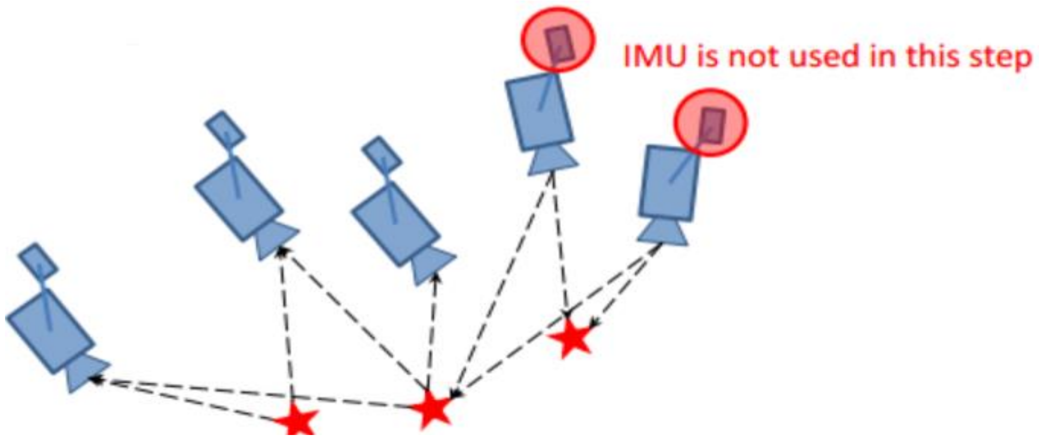
  -stable feature tracking (more than 30 tracked features) and sufficient parallax (more than 20 rotation-compensated pixels)

  -5-point method 、PnP、**global full BA**

  →up-to-scale camera poes + feature positions

  -the **1st camera frame** as the reference frame, i.e., not aligned with gravity

  -**IMU not used in this step**



◆VI Alignment

- Metric scale, world frame
- Gyroscope Bias Calibration

$$\min_{\delta b_w} \sum_{k \in \mathcal{B}} \left\| {\mathbf{q}^{c_0}_{b_{k+1}}}^{-1} \otimes \mathbf{q}^{c_0}_{b_k} \otimes \boldsymbol{\gamma}^{b_k}_{b_{k+1}} \right\|^2$$
$$\boldsymbol{\gamma}^{b_k}_{b_{k+1}} \approx \hat{\boldsymbol{\gamma}}^{b_k}_{b_{k+1}} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{J}^{\gamma}_{b_w}\delta\mathbf{b}_w \end{bmatrix}, \tag{15}$$

$$\hat{\mathbf{z}}^{b_k}_{b_{k+1}} = \begin{bmatrix} \hat{\boldsymbol{\alpha}}^{b_k}_{b_{k+1}} - \mathbf{p}^b_c + \mathbf{R}^{b_k}_{c_0}\mathbf{R}^{c_0}_{b_{k+1}}\mathbf{p}^b_c \\ \hat{\boldsymbol{\beta}}^{b_k}_{b_{k+1}} \end{bmatrix} = \mathbf{H}^{b_k}_{b_{k+1}}\mathcal{X}_I + \mathbf{n}^{b_k}_{b_{k+1}} \tag{18}$$

$$\mathbf{H}^{b_k}_{b_{k+1}} = \begin{bmatrix} -\mathbf{I}\Delta t_k & \mathbf{0} & \frac{1}{2}\mathbf{R}^{b_k}_{c_0}\Delta t_k^2 & \mathbf{R}^{b_k}_{c_0}(\bar{\mathbf{p}}^{c_0}_{c_{k+1}} - \bar{\mathbf{p}}^{c_0}_{c_k}) \\ -\mathbf{I} & \mathbf{R}^{b_k}_{c_0}\mathbf{R}^{c_0}_{b_{k+1}} & \mathbf{R}^{b_k}_{c_0}\Delta t_k & \mathbf{0} \end{bmatrix}$$

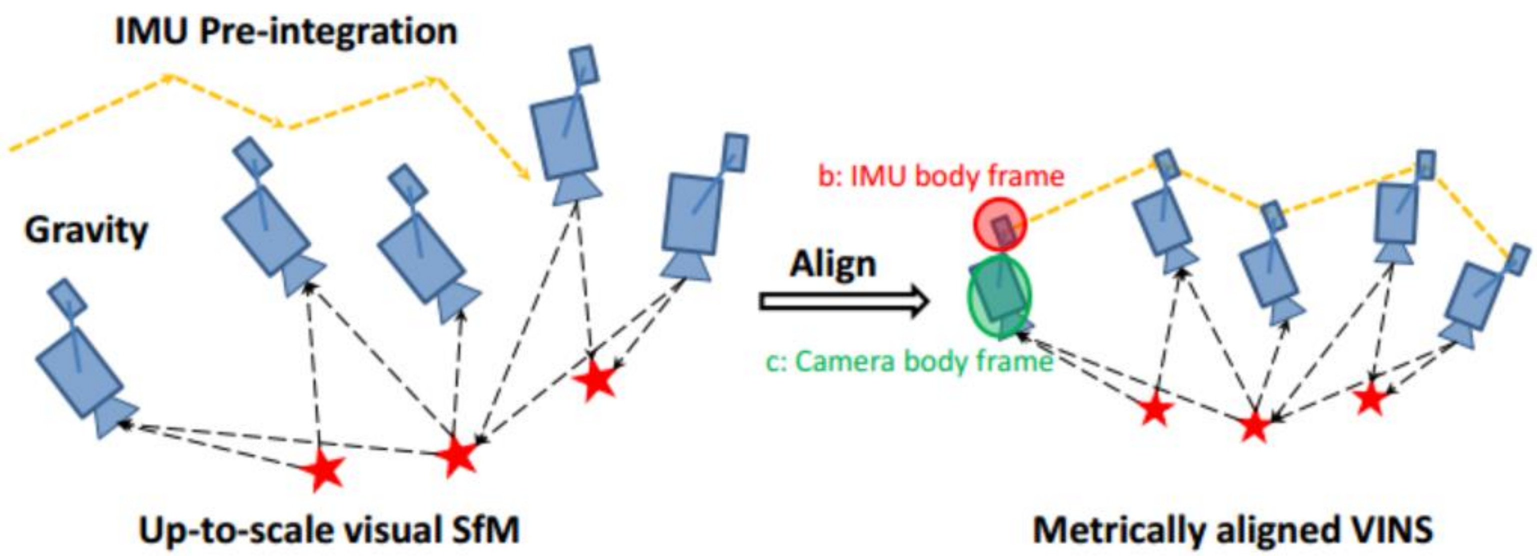
- Velocity, Gravity Vector and Metric Scale Initialization

$$\mathcal{X}_I = \left[\mathbf{v}^{b_0}_{b_0}, \mathbf{v}^{b_1}_{b_1}, \cdots \mathbf{v}^{b_n}_{b_n}, \mathbf{g}^{c_0}, s\right]$$

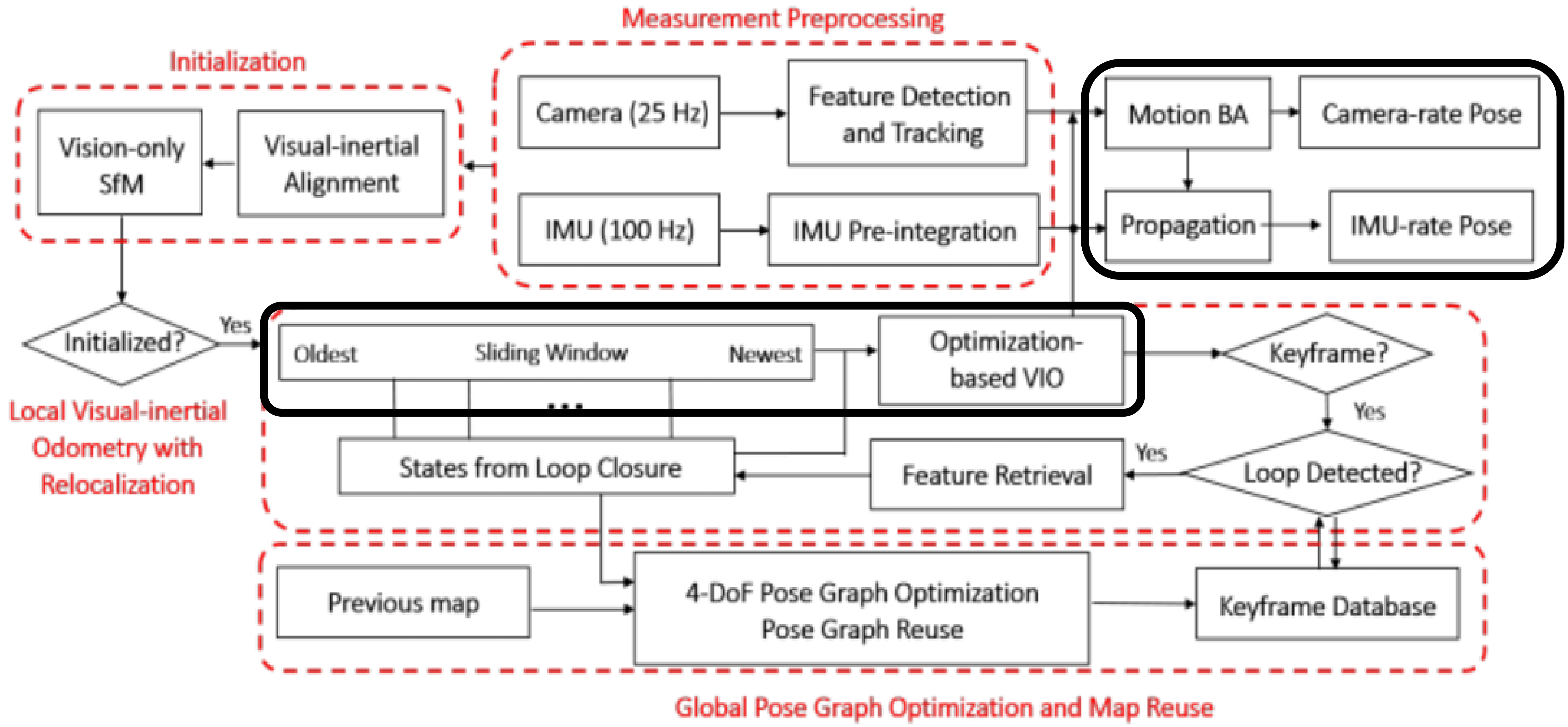
$$\min_{\mathcal{X}_I} \sum_{k \in \mathcal{B}} \left\| \hat{\mathbf{z}}^{b_k}_{b_{k+1}} - \mathbf{H}^{b_k}_{b_{k+1}}\mathcal{X}_I \right\|^2$$

Linear least squre problem

because R and T is known

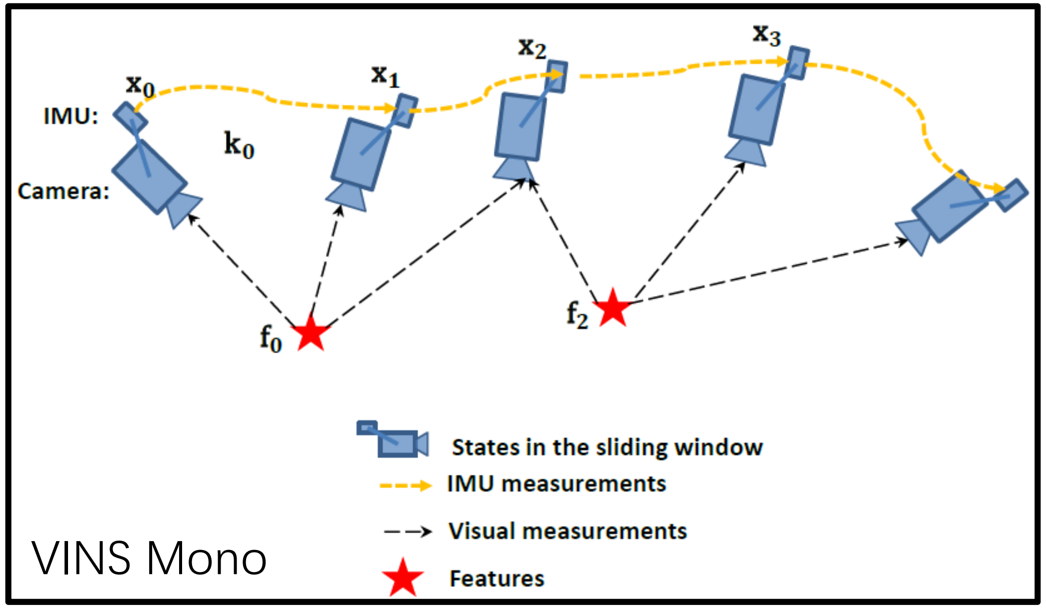


◆VIO



Measurement Preprocessing
Initialization
Vision-only SfM
Visual-inertial Alignment
Camera (25 Hz)
Feature Detection and Tracking
IMU (100 Hz)
IMU Pre-integration
Motion BA
Camera-rate Pose
Propagation
IMU-rate Pose
Initialized?
Yes
Oldest
Sliding Window
Newest
Optimization-based VIO
Keyframe?
Yes
Local Visual-inertial Odometry with Relocalization
States from Loop Closure
Feature Retrieval
Yes
Loop Detected?
Previous map
4-DoF Pose Graph Optimization
Pose Graph Reuse
Keyframe Database
Global Pose Graph Optimization and Map Reuse

◆VIO



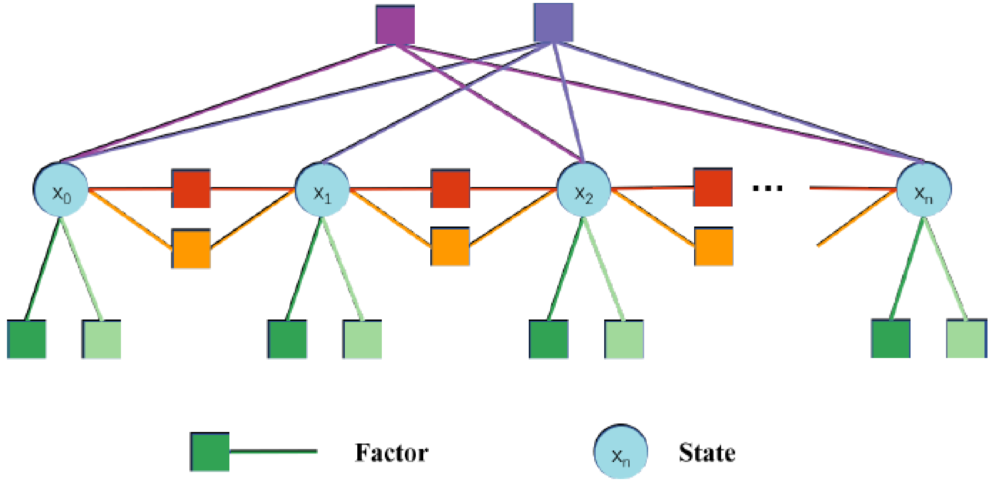
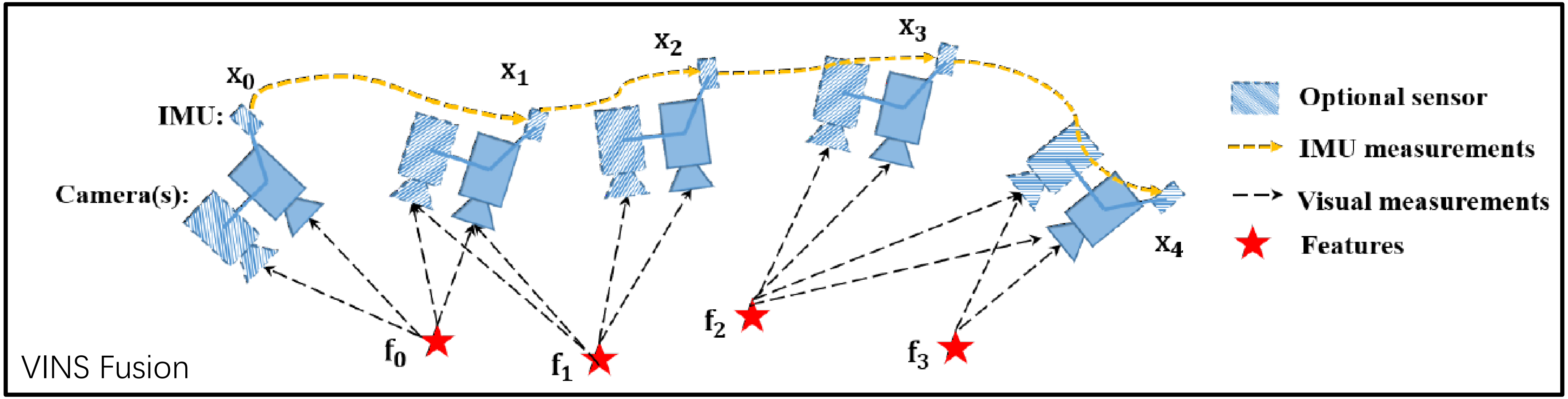


Fig. 2. A graphic illustration of the pose graph. Each node represents states (position, orientation, velocity and so on) at one moment. Each edge represents a factor, which is derived by one measurement. Edges constrain one state, two states or multiple states.



◆VIO

- Nonlinear graph optimization-based, visual-inertial bundle adjustment
- **tightly-coupled**,Sliding window
- The full state vector

$$\mathcal{X} = [\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_n, \mathbf{x}_c^b, \lambda_0, \lambda_1, \cdots \lambda_m]$$
$$\mathbf{x}_k = [\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_g], k \in [0, n]$$
$$\boxed{\mathbf{x}_c^b = [\mathbf{p}_c^b, \mathbf{q}_c^b],}$$

VINS Mono

$$\mathcal{X} = [\mathbf{p}_0, \mathbf{R}_0, \mathbf{p}_1, \mathbf{R}_1, ..., \mathbf{p}_n, \mathbf{R}_n, \mathbf{x}_{cam}, \mathbf{x}_{imu}]$$
$$\mathbf{x}_{cam} = [\lambda_0, \lambda_1, ..., \lambda_l]$$
$$\mathbf{x}_{imu} = [\mathbf{v}_0, \mathbf{b}_{a_0}, \mathbf{b}_{g_0}, \mathbf{v}_1, \mathbf{b}_{a_1}, \mathbf{b}_{g_1}, ..., \mathbf{v}_n, \mathbf{b}_{a_n}, \mathbf{b}_{g_n}]$$

VINS Fusion

n——number of keyframes in sliding window
m——number of features in sliding window

lamda——inverse depth
Xk——IMU state when the kth image is captured

- Minimize residuals from all sensors

$$\min_{\mathcal{X}} \left\{ \boxed{\left\| \mathbf{r}_p - \mathbf{H}_p \mathcal{X} \right\|^2} + \boxed{\sum_{k \in \mathcal{B}} \left\| \mathbf{r}_{\mathcal{B}}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X}) \right\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2} + \boxed{\sum_{(l,j) \in \mathcal{C}} \rho(\left\| \mathbf{r}_{\mathcal{C}}(\hat{\mathbf{z}}_l^{c_j}, \mathcal{X}) \right\|_{\mathbf{P}_l^{c_j}}^2)} \right\},$$

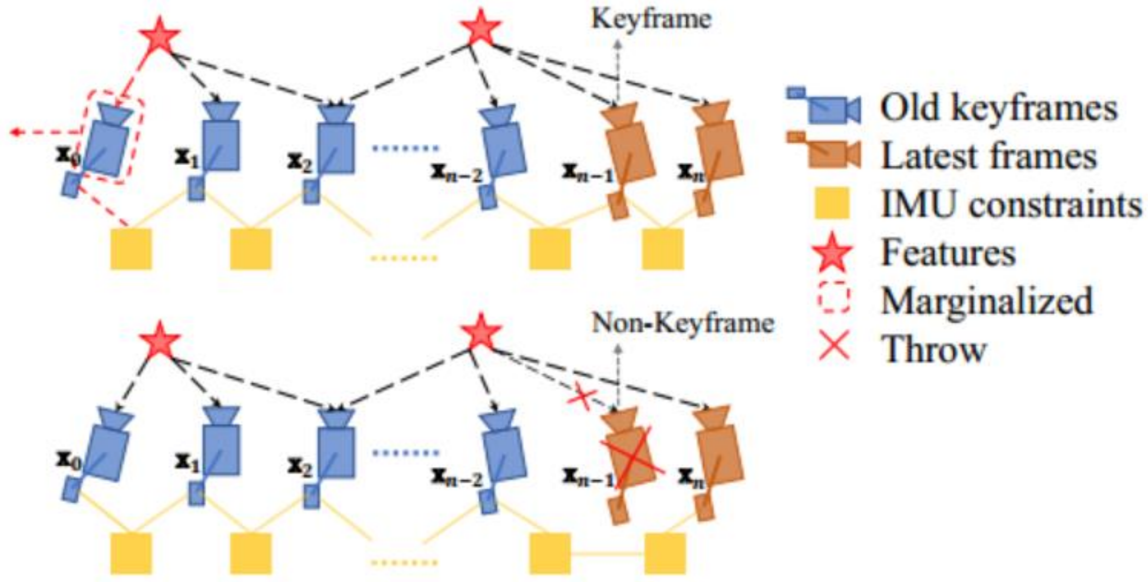
Prior from marginalization
IMU measurement residual
Vision measurement residual

$$\|\mathbf{r}\|_{\Omega}^2 = \mathbf{r}^T \Omega^{-1} \mathbf{r}.$$

Mahalanobis norm

◆Marginalization

- Purpose: bound the computational complexity of optimization-based VIO
- Principles:
  -Add all frames into the sliding window, and remove non-keyframes after the nonlinear optimization
  -keep as many keyframes with sufficient parallax as possible;
  -Maintain matrix sparsity by throwing away visual measurements from nonkeyframes.
- Method: using the Schur complement



$$\begin{bmatrix} \mathbf{H}_{mm} & \mathbf{H}_{mr} \\ \mathbf{H}_{rm} & \mathbf{H}_{rr} \end{bmatrix} \begin{bmatrix} \delta\mathcal{X}_m \\ \delta\mathcal{X}_r \end{bmatrix} = \begin{bmatrix} \mathbf{b}_m \\ \mathbf{b}_r \end{bmatrix}$$

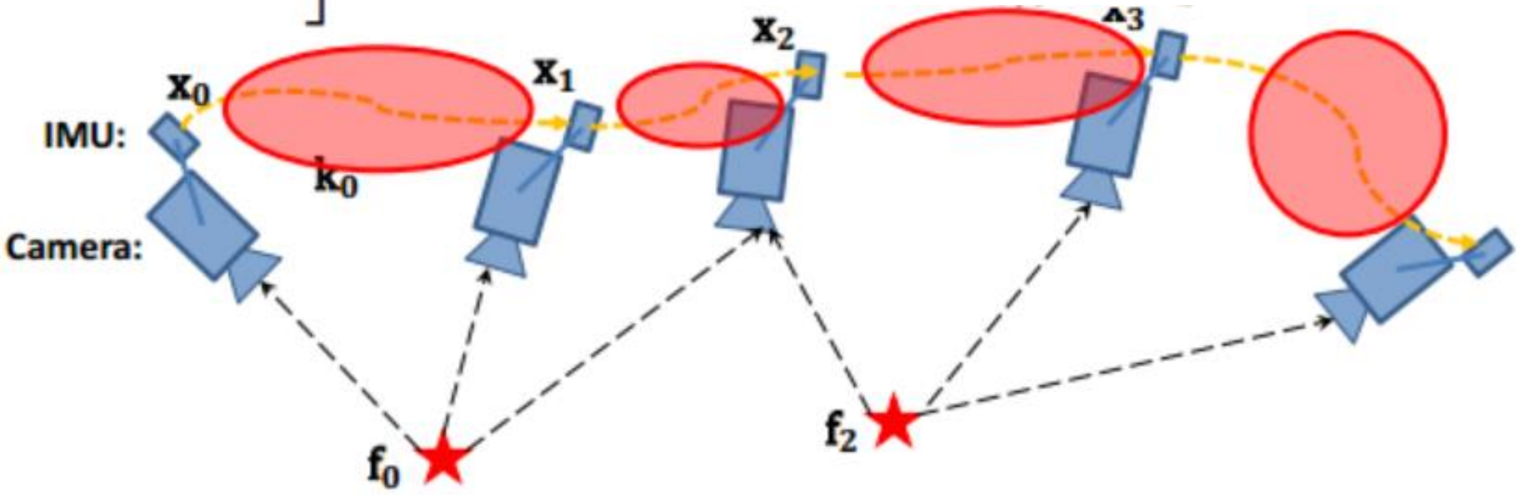
$$\underbrace{(\mathbf{H}_{rr} - \mathbf{H}_{rm}\mathbf{H}_{mm}{}^{-1}\mathbf{H}_{mr})}_{\mathbf{H}_p} \delta\mathcal{X}_r = \underbrace{\mathbf{b}_r - \mathbf{H}_{rm}\mathbf{H}_{mm}{}^{-1}\mathbf{b}_m}_{\mathbf{b}_p}$$

◆IMU Measurement Residual

$$
\mathbf{r}_{\mathcal{B}}(\hat{\mathbf{z}}^{b_k}_{b_{k+1}}, \mathcal{X}) = \begin{bmatrix} \delta\boldsymbol{\alpha}^{b_k}_{b_{k+1}} \\ \delta\boldsymbol{\beta}^{b_k}_{b_{k+1}} \\ \delta\boldsymbol{\theta}^{b_k}_{b_{k+1}} \\ \delta\mathbf{b}_a \\ \delta\mathbf{b}_g \end{bmatrix}
= \begin{bmatrix} \boxed{\mathbf{R}^{b_k}_{w}(\mathbf{p}^w_{b_{k+1}} - \mathbf{p}^w_{b_k} + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 - \mathbf{v}^w_{b_k}\Delta t_k)} - \hat{\boldsymbol{\alpha}}^{b_k}_{b_{k+1}} \\ \boxed{\mathbf{R}^{b_k}_{w}(\mathbf{v}^w_{b_{k+1}} + \mathbf{g}^w \Delta t_k - \mathbf{v}^w_{b_k})} - \hat{\boldsymbol{\beta}}^{b_k}_{b_{k+1}} \\ 2\left[\boxed{\mathbf{q}^{w^{-1}}_{b_k} \otimes \mathbf{q}^w_{b_{k+1}}} \otimes (\hat{\boldsymbol{\gamma}}^{b_k}_{b_{k+1}})^{-1}\right]_{xyz} \\ \mathbf{b}_{ab_{k+1}} - \mathbf{b}_{ab_k} \\ \mathbf{b}_{wb_{k+1}} - \mathbf{b}_{wb_k} \end{bmatrix}
$$

IMU pre-integration “blocks”(observation part)

IMU estimation “blocks”(estimation part)



◆Visual Measurement Residual

$$\mathbf{r}_C(\hat{\mathbf{z}}_l^{c_j}, \mathcal{X}) = [\mathbf{b}_1 \quad \mathbf{b}_2]^T \cdot \left(\hat{\mathcal{P}}_l^{c_j} - \frac{\mathcal{P}_l^{c_j}}{\|\mathcal{P}_l^{c_j}\|}\right)$$

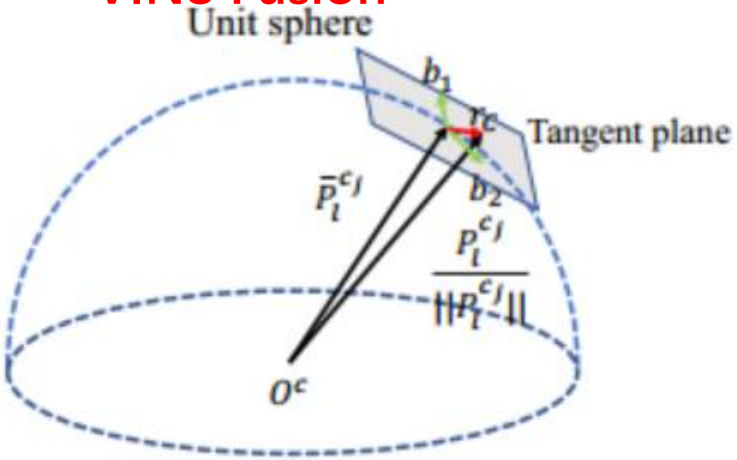
$\hat{\mathcal{P}}_l^{c_j} = \pi^{-1}\left(\left[\hat{u}_l^{c_j}\right]\right)$

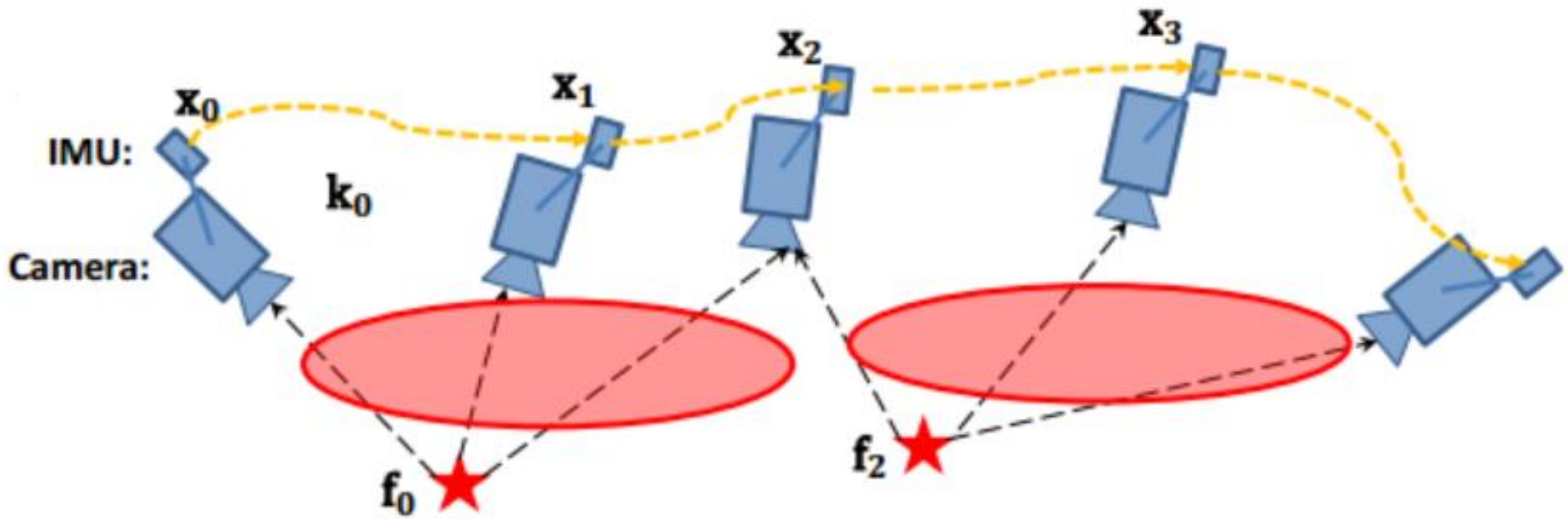
$\boxed{\mathcal{P}_l^{c_j}} =$

- **Reprojection error**
- Inverse depth model
- del

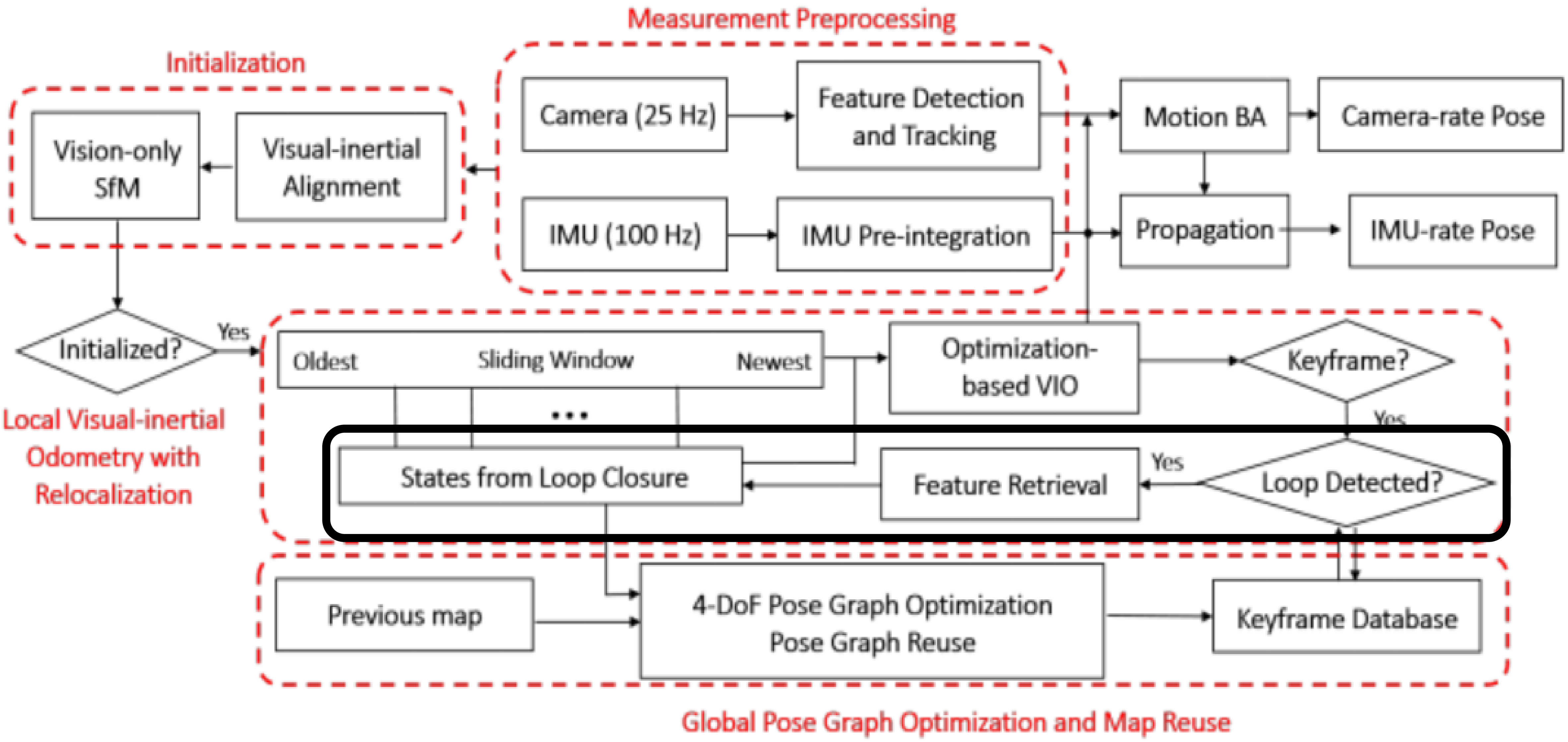
This factor is universal for both left camera and right camera. We can project a feature from the left image to the left image in temporal space, also we can project a feature from the left image to the right image in spatial space.

VINS Fusion



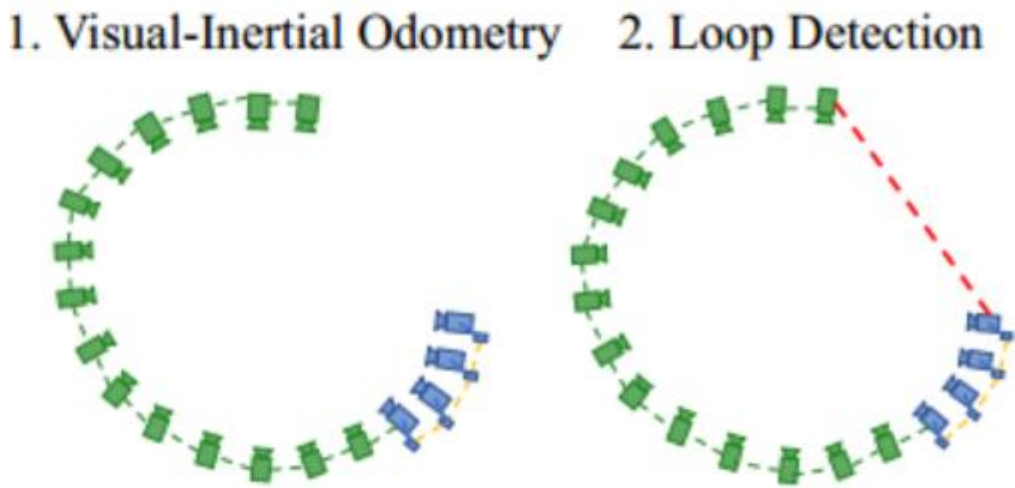


◆RELOCALIZATION



◆Loop Detection

- Features described by the BRIEF descriptor
  - -Harris corners used in VIO
  - -Additional 500 more FAST corners
- DBoW2 returns loop closure candidates



consistency check. We keep all BRIEF descriptors for feature retrieving, but discard the raw image to reduce memory consumption.

We note that our monocular VIO is able to render roll and pitch angles observable. As such, we do not need to rely on rotation-invariant features, such as the ORB feature used in ORB SLAM [4].

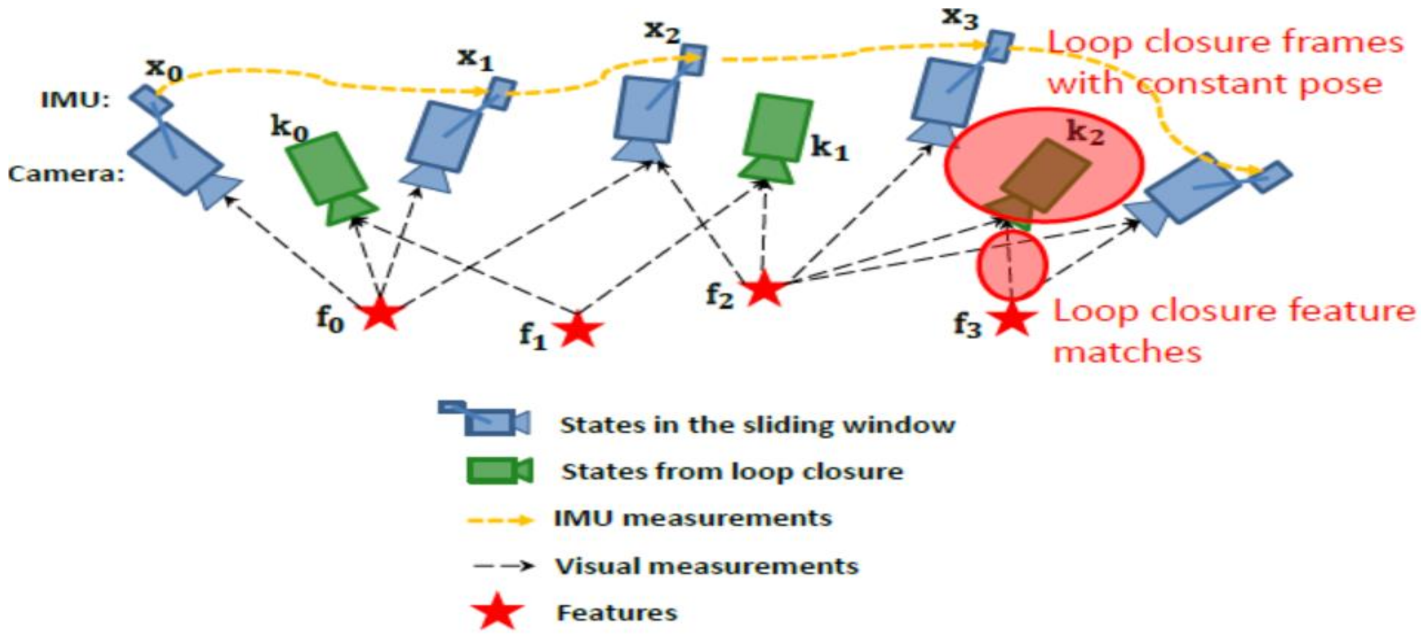
◆Feature Retrieval

- BRIEF descriptor matching
- 2D-2D: fundamental matrix test with RANSAC
- 3D-2D: PnP test with RANSAC
- the number of inliers beyond a certain threshold, then loop closure OK

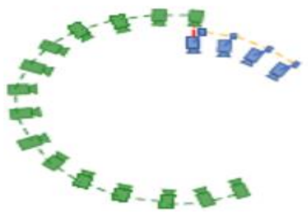
◆Tightly-Coupled Relocalization

$$\min_{\mathcal{X}} \left\{ \boxed{\left\|\mathbf{r}_p - \mathbf{H}_p\mathcal{X}\right\|^2 + \sum_{k\in\mathcal{B}} \left\|\mathbf{r}_{\mathcal{B}}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X})\right\|^2_{\mathbf{P}^{b_k}_{b_{k+1}}} + \sum_{(l,j)\in\mathcal{C}} \rho\left(\left\|\mathbf{r}_{\mathcal{C}}(\hat{\mathbf{z}}_l^{c_j},\mathcal{X})\right\|^2_{\mathbf{P}_l^{c_j}}\right)} + \sum_{(l,v)\in\mathcal{L}} \rho\left(\left\|\mathbf{r}_{\mathcal{C}}(\hat{\mathbf{z}}_l^{v}, \mathcal{X}, \hat{\mathbf{q}}_v^{w}, \hat{\mathbf{p}}_v^{w})\right\|^2_{\mathbf{P}_l^{c_v}}\right) \right\},$$

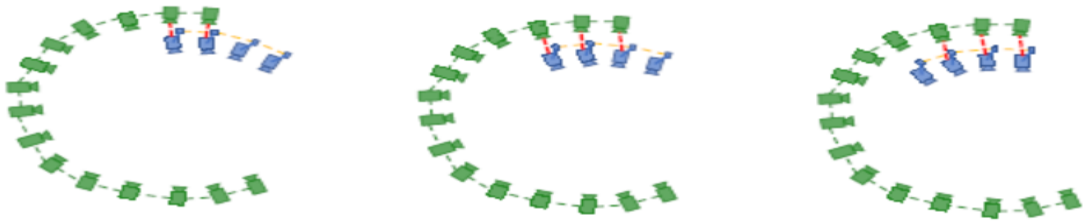
- VIO Residuals
- Loop closure vision measurement residual
- Poses of loop closure frames are constant
- multi-view constraints for relocalization



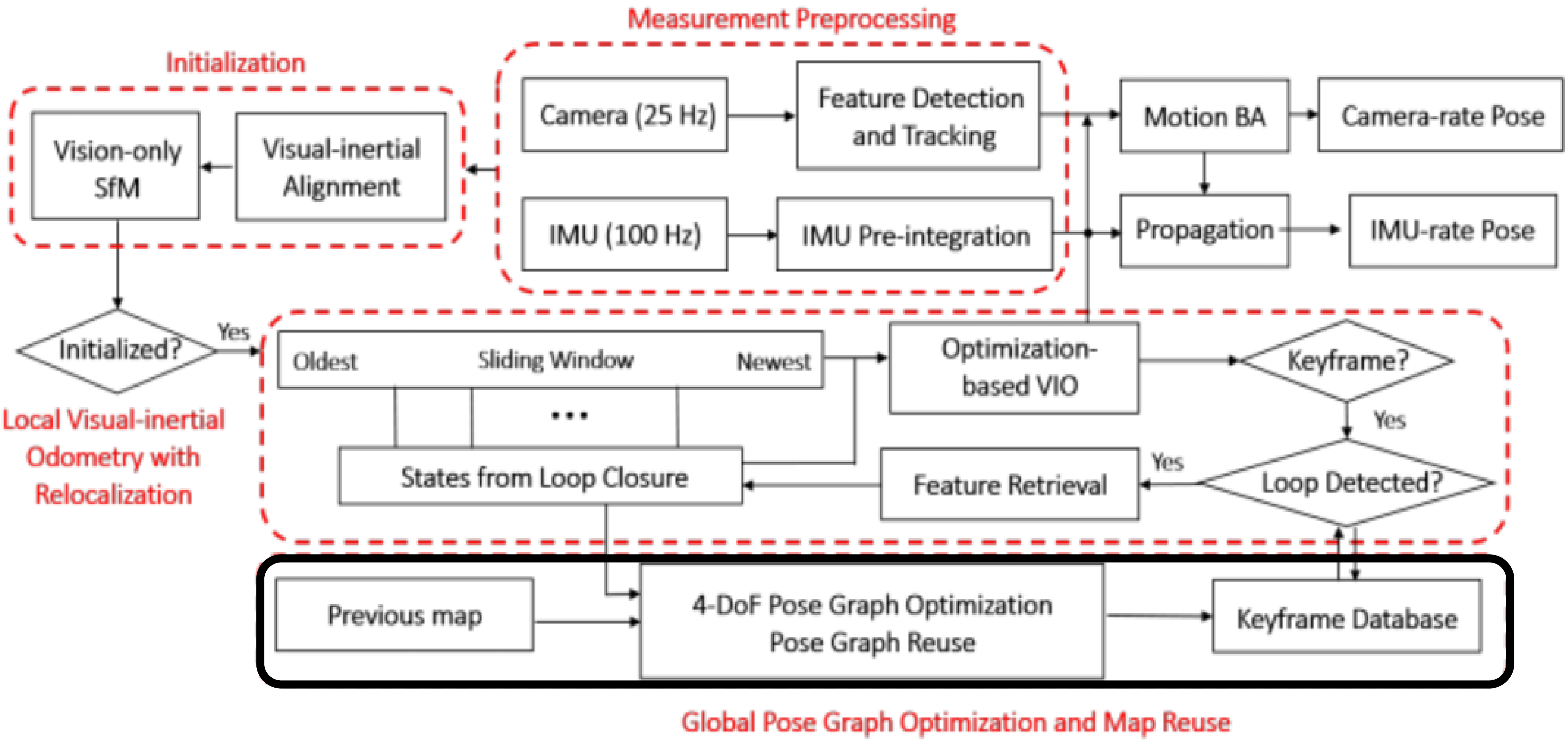
3. Relocalization



4. Relocalization with Multiple Constraints



◆GLOBAL POSE GRAPH OPTIMIZATION



◆GLOBAL POSE GRAPH OPTIMIZATION

- 4-DOF pose graph optimization(x, y, z and yaw angle)

  -Roll and pitch are observable from VIO

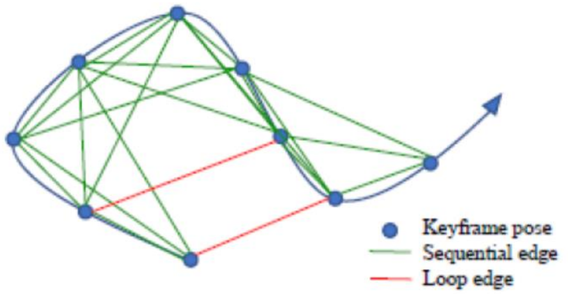
- Relocalization and pose graph optimization run in different threads and different rate
- Adding Keyframes into the Pose Graph

  -When a keyframe is marginalized out from the sliding window

  -two types of edges:

  1) Sequential Edge

  2) Loop Closure Edge



- 4-DOF relative pose residual

  -obtained directly from VIO

$$\mathbf{r}_{i,j}(\mathbf{p}_i^w, \psi_i, \mathbf{p}_j^w, \psi_j) = \begin{bmatrix} \mathbf{R}(\boxed{\hat{\phi}_i, \hat{\theta}_i}, \psi_i)^{-1}(\mathbf{p}_j^w - \mathbf{p}_i^w) - \hat{\mathbf{p}}_{ij}^i \\ \psi_j - \psi_i - \hat{\psi}_{ij} \end{bmatrix}, \tag{28}$$

◆GLOBAL POSE GRAPH OPTIMIZATION

- minimizing the following cost function

$$\min_{\mathbf{p},\psi}\left\{\sum_{(i,j)\in\mathcal{S}}\|\mathbf{r}_{i,j}\|^2+\sum_{(i,j)\in\mathcal{L}}\boxed{\rho}(\|\mathbf{r}_{i,j}\|^2)\right\}$$

Sequential edges

Loop closure edges

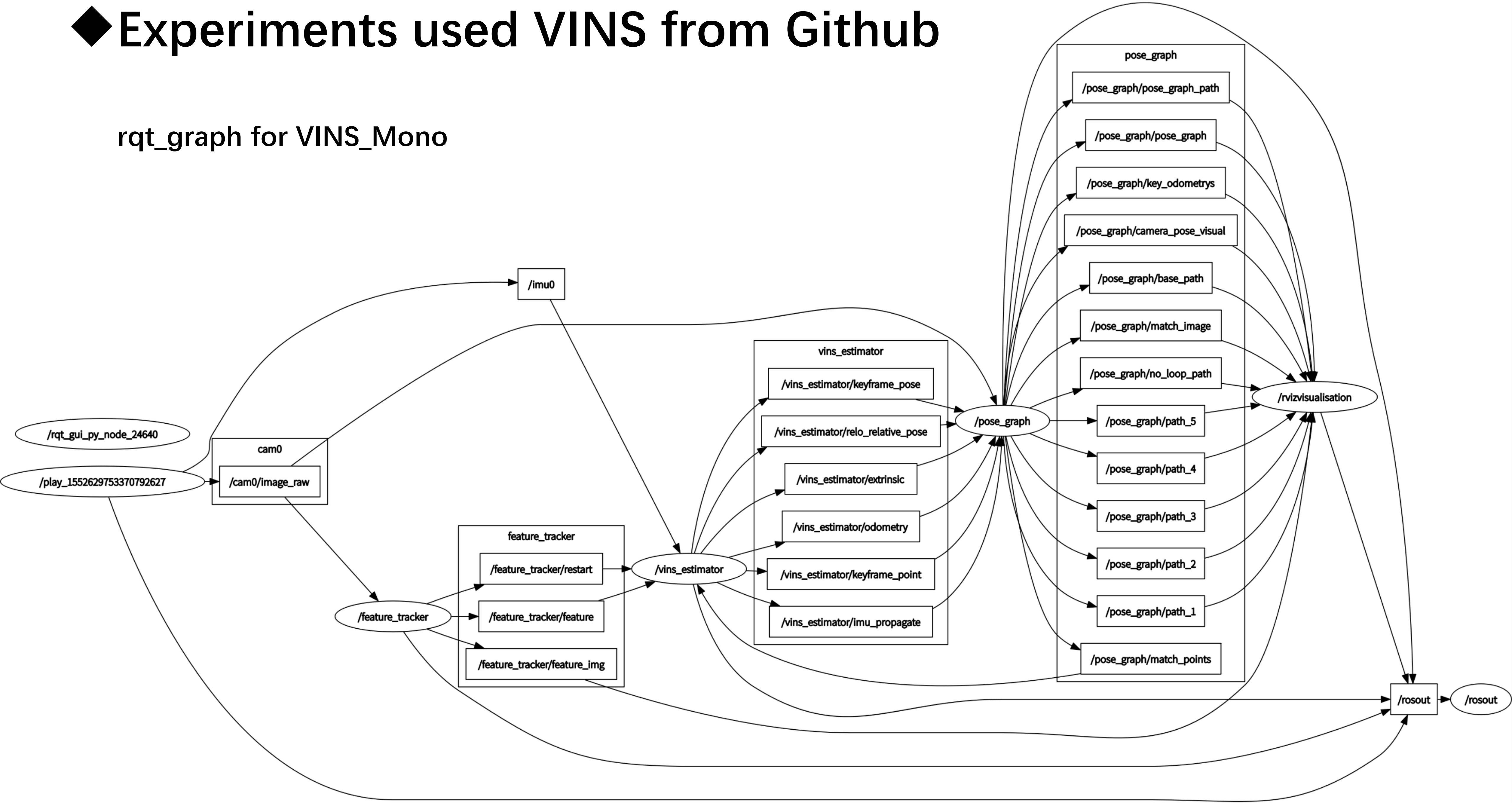
Huber norm, used to further reduce the impact of any possible wrong loops

- Pose Graph Management

of the system in the long run. To this end, we implement a downsample process to maintain the pose graph database to a limited size. All keyframes with loop closure constraints will be kept, while other keyframes that are either too close or have very similar orientations to its neighbors may be removed. The probability of a keyframe being removed is proportional to its spatial density to its neighbors.

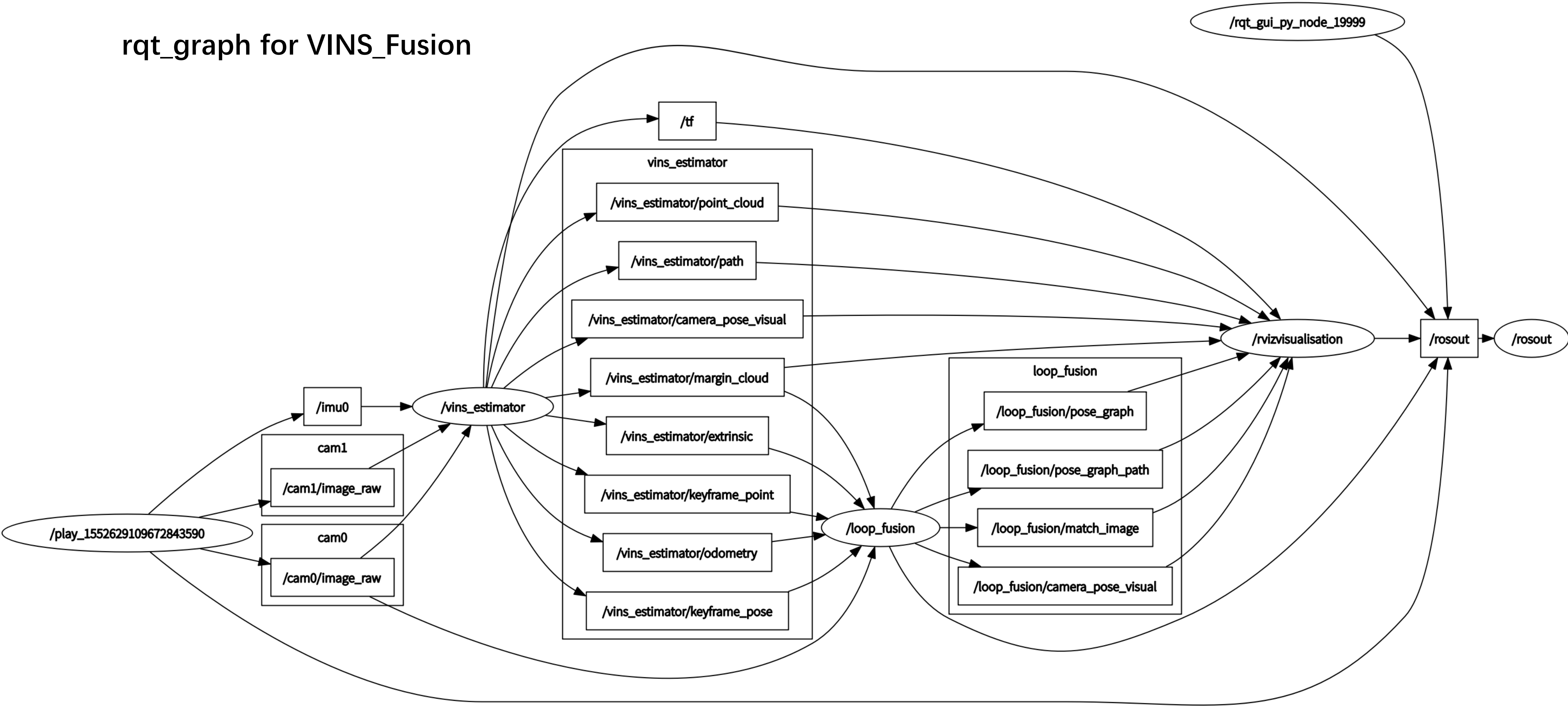
◆Experiments used VINS from Github

rqt_graph for VINS_Mono



◆Experiments used VINS from Github

rqt_graph for VINS_Fusion



Thanks!