ORBSLAM3-2

Bit

Bitterengsci

Outline

- Three Literature between ORBSLAM2 and ORBSLAM3
- [ORBSLAM-VI] Visual-Inertial Monocular SLAM with Map Reuse, IEEE RAL, 2017
- [ORBSLAM-Atlas] ORBSLAM-Atlas: a robust and accurate multi-map system, IROS 2019
- [IMU-Initialization] Inertial-Only Optimization for Visual-Inertial Initialization, ICRA 2020

ORBSLAM-VI: Overview

- 3 threads: tracking, local mapping, loop closing
- work on large-scale environments
  - build a covisibility graph to recover local maps for tracking and mapping
  - lightweight pose optimization for loop closure
- allow localization-only mode (mapping & loop closing disabled)
- IMU-Initialization

ORBSLAM-VI:

3 threads: tracking, local mapping, loop closing

work on large-scale environments

build a covisibility graph to recover local maps for tracking and mapping

lightweight pose optimization for loop closure

allow localization-only mode (mapping & loop closing disabled) less CPU-intensive

IMU-Initialization

ORBSLAM-VI: Tracking

- In charge of tracking sensor pose, velocity, IMU biases at frame rate
  - 1. reliable camera pose prediction
  - 2. map points in the local map are projected and matched with keypoints on the frame
  - 3. Optimize the cost (feature reprojection error of all matched points + IMU error)

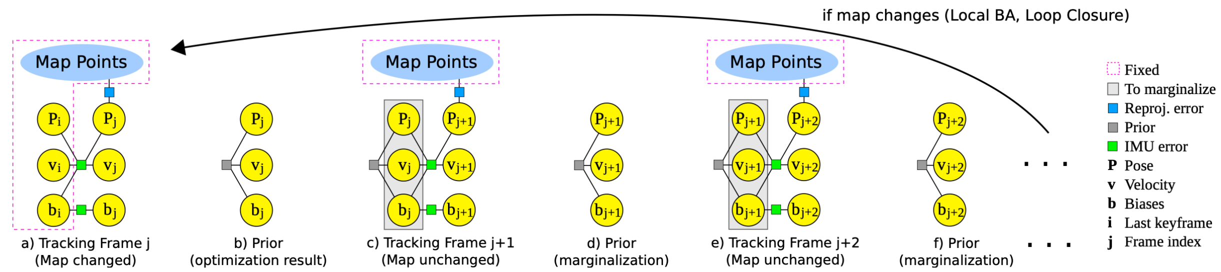


Fig. 2. Evolution of the optimization in the Tracking thread. (a) We start optimizing the frame j linked by an IMU constraint to last keyframe i. (b) The result of the optimization (estimation and Hessian matrix) serves as prior for next optimization. (c) When tracking next frame $j+1$, both frames j and $j+1$ are jointly optimized, being linked by an IMU constraint, and having frame j the prior from previous optimization. (d) At the end of the optimization, the frame j is marginalized out and the result serves as prior for following optimization. (e-f) This process is repeated until there is a map update from the Local Mapping or Loop Closing thread. In such case the optimization links the current frame to last keyframe discarding the prior, which is not valid after the map change.

Case 1: tracking performed just after a map update
Case 2: no map update (localization-only mode always in this case)

In charge of tracking sensor pose, velocity, IMU biases at frame rate

1.reliable camera pose prediction
2.map points in the local map are projected and matched with keypoints on the frame
3.Optimize the cost (feature reprojection error of all matched points + IMU error)

    2 cases (whether map being updated by the local mapping or the loop closing thread)
    Case 1: tracking performed just after a map update
    Case 2: no map update (localization-only mode always in this case)

ORBSLAM-VI: Tracking (Continued)

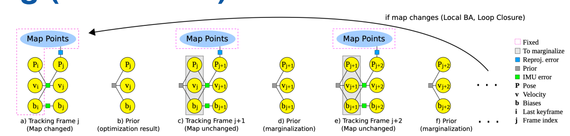


Fig. 2. Evolution of the optimization in the Tracking thread. (a) We start optimizing the frame j linked by an IMU constraint to last keyframe i. (b) The result of the optimization (estimation and Hessian matrix) serves as prior for next optimization. (c) When tracking next frame $j+1$, both frames j and $j+1$ are jointly optimized, being linked by an IMU constraint, and having frame j the prior from previous optimization. (d) At the end of the optimization, the frame j is marginalized out and the result serves as prior for following optimization. (e-f) This process is repeated until there is a map update from the Local Mapping or Loop Closing thread. In such case the optimization links the current frame to last keyframe discarding the prior, which is not valid after the map change.

- Case 1: tracking performed just after a map update

$$\theta = \left\{ \mathbf{R}^j_{\mathtt{WB}}, {}_{\mathtt{W}}\mathbf{p}^j_{\mathtt{B}}, {}_{\mathtt{W}}\mathbf{v}^j_{\mathtt{B}}, \mathbf{b}^j_g, \mathbf{b}^j_a \right\}$$

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

- Case 2: no map update (localization-only mode always in this case)

$$\theta = \left\{ \mathbf{R}^j_{\mathtt{WB}}, \mathbf{p}^j_{\mathtt{W}}, \mathbf{v}^j_{\mathtt{W}}, \mathbf{b}^j_g, \mathbf{b}^j_a, \mathbf{R}^{j+1}_{\mathtt{WB}}, \mathbf{p}^{j+1}_{\mathtt{W}}, \mathbf{v}^{j+1}_{\mathtt{W}}, \mathbf{b}^{j+1}_g, \mathbf{b}^{j+1}_a \right\}$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \Big(\sum_k \mathbf{E}_{\text{proj}}(k, j+1) + \mathbf{E}_{\text{IMU}}(j, j+1) + \mathbf{E}_{\text{prior}}(j) \Big) \quad (7)$$

- Solved by Gauss-Newton algorithm in g2o
- Resulting estimation and Hessian as prior for next optimization

Solved by Gauss-Newton algorithm in g2o
Resulting estimation and Hessian as prior for next optimization

SideNote: ORBSLAM-VI: Tracking (Appendix)

$$\mathbf{E}_{\text{proj}}(k,j) = \rho\left(\left(\mathbf{x}^k - \pi(\mathbf{X}_{\mathtt{C}}^k)\right)^T \boldsymbol{\Sigma}_{\boldsymbol{k}} \left(\mathbf{x}^k - \pi(\mathbf{X}_{\mathtt{C}}^k)\right)\right)$$

$$\mathbf{X}_{\mathtt{C}}^k = \mathbf{R}_{\mathtt{CB}} \mathbf{R}_{\mathtt{BW}}^j \left(\mathbf{X}_{\mathtt{W}}^k - {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^j\right) + {}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}$$

$$\begin{aligned}
\mathbf{E}_{\text{IMU}}(i,j) &= \rho\left(\left[\mathbf{e}_R^T\, \mathbf{e}_v^T\, \mathbf{e}_p^T\right] \boldsymbol{\Sigma}_I \left[\mathbf{e}_R^T\, \mathbf{e}_v^T\, \mathbf{e}_p^T\right]^T\right) \\
&\quad + \rho\left(\mathbf{e}_b^T \boldsymbol{\Sigma}_R \mathbf{e}_b\right) \\
\mathbf{e}_R &= \mathrm{Log}\left(\left(\Delta\mathbf{R}_{ij} \mathrm{Exp}\left(\mathbf{J}_{\Delta R}^g \mathbf{b}_g^j\right)\right)^T \mathbf{R}_{\mathtt{BW}}^i \mathbf{R}_{\mathtt{WB}}^j\right) \\
\mathbf{e}_v &= \mathbf{R}_{\mathtt{BW}}^i \left({}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^j - {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^i - \mathbf{g}_{\mathtt{W}} \Delta t_{ij}\right) \\
&\quad - \left(\Delta\mathbf{v}_{ij} + \mathbf{J}_{\Delta v}^g \mathbf{b}_g^j + \mathbf{J}_{\Delta v}^a \mathbf{b}_a^j\right) \\
\mathbf{e}_p &= \mathbf{R}_{\mathtt{BW}}^i \left({}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^j - {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^i - {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^i \Delta t_{ij} - \frac{1}{2}\mathbf{g}_{\mathtt{W}} \Delta t_{ij}^2\right) \\
&\quad - \left(\Delta\mathbf{p}_{ij} + \mathbf{J}_{\Delta p}^g \mathbf{b}_g^j + \mathbf{J}_{\Delta p}^a \mathbf{b}_a^j\right) \\
\mathbf{e}_b &= \mathbf{b}^j - \mathbf{b}^i
\end{aligned}$$

$$\mathbf{E}_{\text{prior}}(j) = \rho\left(\left[\mathbf{e}_R^T\, \mathbf{e}_v^T\, \mathbf{e}_p^T\, \mathbf{e}_b^T\right] \boldsymbol{\Sigma}_p \left[\mathbf{e}_R^T\, \mathbf{e}_v^T\, \mathbf{e}_p^T\, \mathbf{e}_b^T\right]^T\right)$$

$$\mathbf{e}_R = \mathrm{Log}\left(\bar{\mathbf{R}}_{\mathtt{BW}}^j \mathbf{R}_{\mathtt{WB}}^j\right) \qquad \mathbf{e}_v = {}_{\mathtt{W}}\bar{\mathbf{v}}_{\mathtt{B}}^j - {}_{\mathtt{W}}\mathbf{v}_{\mathtt{B}}^j$$

$$\mathbf{e}_p = {}_{\mathtt{W}}\bar{\mathbf{p}}_{\mathtt{B}}^j - {}_{\mathtt{W}}\mathbf{p}_{\mathtt{B}}^j \qquad \mathbf{e}_b = \bar{\mathbf{b}}^j - \mathbf{b}^j$$

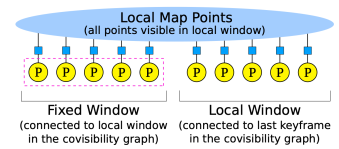
x_k: keypoint location in the image

X_kw: map points in the world coordinate
Σk: information matrix associated to the keypoint scale
ΣI: info matrix of the preintegration   ΣR: bias random walk
ϱ: Huber robust cost

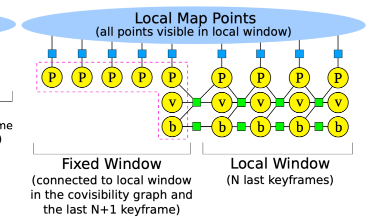
State with bar: estimated states from previous optimization, Σp: Hession matrix from previous optimization

After this optimization (Fig. 2d), frame j is marginalized out [5]. This optimization linking two consecutive frames and using a prior is repeated (Fig. 2e-f) until a map change, when the prior will be no longer valid and the tracking will link again the current frame to the last keyframe (Fig. 2a).

ORBSLAM-VI: Local Mapping



ORB-SLAM's Local BA



Visual-Inertial ORB-SLAM's Local BA

- responsible for:
  - 1. Local BA after a new keyframe insertion
    - Cost = reprojection error + IMU error
    - Similar to ORB: local window + fixed window
    - Last keyframe N+1 always in fixed window (it constrains IMU states)
  - 2.Keyframe management
    - Old ORBSLAM keyframe discarding policy problematic
      - the temporal difference between keyframes larger, the IMU info weaker
    - so, allowing the map to discard redundant keyframes only when:
      - two consecutive keyframes (local window) of local BA differ no more than 0.5s
      - two consecutive keyframes of full BA (after a loop closure or at any time to refine a map) differ no more than 3s

1.**Local BA after a new keyframe insertion**

cost = reprojection error + IMU error (similar to tracking part)
all points seen by last N keyframes and themselves (local window)
“related” keyframes fixed but contribute to total cost (fixed window)
Last keyframe N+1 always in fixed window (it constrains IMU states)

2.**Keyframe management**

ORBSLAM policy discards redundant keyframes (map size does now grow if localizing in a well mapped area)
problematic if IMU info involved, since it constrains the motion of consecutive keyframes (the longer the temporal difference between consecutive keyframes, the weaker info IMU provides)

ORBSLAM-VI: Loop Closing

- 1. Match a recent keyframe with a past keyframe (place recognition module)
- 2. Validated by computing rigid-body transformation that aligns matched points between keyframes
- 3. Pose Optimization to reduce accumulated error in the trajectory
  - ignore structure to survive complexity
  - 6 DoF, scale observable
  - excl. IMU info (velocity, biases)
- 4. fullBA (in a parallel thread) to optimize all states (incl. IMU info)

To reduce drift accumulation during exploration (when returned to an already mapped area)

1. Match a recent keyframe with a past keyframe (place recognition module)
2. Validated by computing rigid-body transformation that aligns matched points between keyframes ??? [22]
3. Pose Optimization to reduce accumulated error in the trajectory
    ignore structure to survive complexity
    6 DoF, scale observable
    excl. IMU info (velocity, biases) $\rightarrow$ correct velocities by rotating them according to the corrected orientation of the associated keyframe
4. fullBA (in a parallel thread) to optimize all states (incl. IMU info)

ORBSLAM-VI: IMU Initialization

- 1.Gyroscope Bias Estimation (from the known orientation of 2 consecutive keyframes)

$$\underset{\mathbf{b}_g}{\operatorname{argmin}} \sum_{i=1}^{N-1} \left\| \operatorname{Log}\left(\left(\Delta\mathbf{R}_{i,i+1} \operatorname{Exp}\left(\mathbf{J}^g_{\Delta R}\mathbf{b}_g\right)\right)^T \mathbf{R}^{i+1}_{\mathtt{BW}} \mathbf{R}^i_{\mathtt{WB}} \right) \right\|^2$$

- 2.Scale and Gravity Approximation (excl. accelerometer bias)

$$s\,{}_{\mathtt{W}}\mathbf{p}^{i+1}_{\mathtt{C}} = s\,{}_{\mathtt{W}}\mathbf{p}^{i}_{\mathtt{C}} + {}_{\mathtt{W}}\mathbf{v}^i_{\mathtt{B}}\Delta t_{i,i+1} + \frac{1}{2}\mathbf{g}_{\mathtt{W}}\Delta t^2_{i,i+1} + \mathbf{R}^i_{\mathtt{WB}}\Delta\mathbf{p}_{i,i+1} + \left(\mathbf{R}^i_{\mathtt{WC}} - \mathbf{R}^{i+1}_{\mathtt{WC}}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}$$

$$\begin{bmatrix}\boldsymbol{\lambda}(i) & \boldsymbol{\beta}(i)\end{bmatrix}\begin{bmatrix} s \\ \mathbf{g}_{\mathtt{W}}\end{bmatrix} = \boldsymbol{\gamma}(i)$$

$$\begin{aligned}
\boldsymbol{\lambda}(i) &= \left({}_{\mathtt{W}}\mathbf{p}^2_{\mathtt{C}} - {}_{\mathtt{W}}\mathbf{p}^1_{\mathtt{C}}\right)\Delta t_{23} - \left({}_{\mathtt{W}}\mathbf{p}^3_{\mathtt{C}} - {}_{\mathtt{W}}\mathbf{p}^2_{\mathtt{C}}\right)\Delta t_{12} \\
\boldsymbol{\beta}(i) &= \frac{1}{2}\mathbf{I}_{3\times 3}\left(\Delta t^2_{12}\Delta t_{23} + \Delta t^2_{23}\Delta t_{12}\right) \\
\boldsymbol{\gamma}(i) &= \left(\mathbf{R}^2_{\mathtt{WC}} - \mathbf{R}^1_{\mathtt{WC}}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{23} - \left(\mathbf{R}^3_{\mathtt{WC}} - \mathbf{R}^2_{\mathtt{WC}}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{12} \\
&\quad + \mathbf{R}^2_{\mathtt{WB}}\Delta\mathbf{p}_{23}\Delta t_{12} + \mathbf{R}^1_{\mathtt{WB}}\Delta\mathbf{v}_{12}\Delta t_{12}\Delta t_{23} \\
&\quad - \mathbf{R}^1_{\mathtt{WB}}\Delta\mathbf{p}_{12}\Delta t_{23}
\end{aligned} \tag{13}$$

- 3.Accelerometer Bias Estimation, Scale and Gravity Direction Refinement

$$\mathbf{g}_{\mathtt{W}} = \mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G$$

$$\mathbf{g}_{\mathtt{W}} = \mathbf{R}_{\mathtt{WI}}\operatorname{Exp}(\boldsymbol{\delta\theta})\,\hat{\mathbf{g}}_{\mathtt{I}}\,G$$

$$\boldsymbol{\delta\theta} = \left[\boldsymbol{\delta\theta}^T_{\boldsymbol{xy}}\ 0\right]^T, \quad \boldsymbol{\delta\theta}_{\boldsymbol{xy}} = \left[\delta\theta_x\ \delta\theta_y\right]^T$$

$$\mathbf{g}_{\mathtt{W}} \approx \mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G - \mathbf{R}_{\mathtt{WI}}\left(\hat{\mathbf{g}}_{\mathtt{I}}\right)_{\times}G\,\boldsymbol{\delta\theta}$$

$$\begin{aligned}
s\,{}_{\mathtt{W}}\mathbf{p}^{i+1}_{\mathtt{C}} &= s\,{}_{\mathtt{W}}\mathbf{p}^{i}_{\mathtt{C}} + {}_{\mathtt{W}}\mathbf{v}^i_{\mathtt{B}}\Delta t_{i,i+1} - \frac{1}{2}\mathbf{R}_{\mathtt{WI}}\left(\hat{\mathbf{g}}_{\mathtt{I}}\right)_{\times}G\Delta t^2_{i,i+1}\,\boldsymbol{\delta\theta} \\
&\quad + \mathbf{R}^i_{\mathtt{WB}}\left(\Delta\mathbf{p}_{i,i+1} + \mathbf{J}^a_{\Delta p}\mathbf{b}_a\right) + \left(\mathbf{R}^i_{\mathtt{WC}} - \mathbf{R}^{i+1}_{\mathtt{WC}}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}} \\
&\quad + \frac{1}{2}\mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G\Delta t^2_{i,i+1}
\end{aligned} \tag{18}$$

$$\begin{bmatrix}\boldsymbol{\lambda}(i) & \boldsymbol{\phi}(i) & \boldsymbol{\zeta}(i)\end{bmatrix}\begin{bmatrix} s \\ \boldsymbol{\delta\theta}_{\boldsymbol{xy}} \\ \mathbf{b}_a\end{bmatrix} = \boldsymbol{\psi}(i)$$

- 4.Velocity Estimation
  - Velocities for all keyframes can be computed.
  - Velocities for the most recent keyframe can be computed.
- Bias Reinitialization after Relocalization (use 20 consecutive frames localized with only vision)
  - Gyroscope bias by solving -> (1)
  - Accelerometer bias by solving -> (3)

$$\begin{aligned}
\boldsymbol{\phi}(i) &= \left[\frac{1}{2}\mathbf{R}_{\mathtt{WI}}\left(\hat{\mathbf{g}}_{\mathtt{I}}\right)_{\times}G\left(\Delta t^2_{12}\Delta t_{23} + \Delta t^2_{23}\Delta t_{12}\right)\right]_{(:,1:2)} \\
\boldsymbol{\zeta}(i) &= \mathbf{R}^2_{\mathtt{WB}}\mathbf{J}^a_{\Delta p_{23}}\Delta t_{12} + \mathbf{R}^1_{\mathtt{WB}}\mathbf{J}^a_{\Delta v_{23}}\Delta t_{12}\Delta t_{23} \\
&\quad - \mathbf{R}^1_{\mathtt{WB}}\mathbf{J}^a_{\Delta p_{12}}\Delta t_{23} \\
\boldsymbol{\psi}(i) &= \left(\mathbf{R}^2_{\mathtt{WC}} - \mathbf{R}^1_{\mathtt{WC}}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{23} - \left(\mathbf{R}^3_{\mathtt{WC}} - \mathbf{R}^2_{\mathtt{WC}}\right){}_{\mathtt{C}}\mathbf{p}_{\mathtt{B}}\Delta t_{12} \\
&\quad + \mathbf{R}^2_{\mathtt{WB}}\Delta\mathbf{p}_{23}\Delta t_{12} + \mathbf{R}^1_{\mathtt{WB}}\Delta\mathbf{v}_{12}\Delta t_{12}\Delta t_{23} \\
&\quad - \mathbf{R}^1_{\mathtt{WB}}\Delta\mathbf{p}_{12}\Delta t_{23} + \frac{1}{2}\mathbf{R}_{\mathtt{WI}}\,\hat{\mathbf{g}}_{\mathtt{I}}\,G\Delta t^2_{ij}
\end{aligned} \tag{20}$$

Given: a set of keyframes processed by a Mono SLAM

Require:

- mono SLAM running for a few seconds, sensor performs a motion(有运动, make all variable observable)
- any two consecutive keyframes close in time (to reduce IMU noise integration)

Method to compute an initial estimation for a VI full BA of the scale, gravity direction, velocity and IMU biases

1.Gyroscope Bias Estimation

R_WB = R_WC (orientation computed from ORBSLAM) * R_CB (calibration)

ΔRi, i+1 gyroscope integration between 2 consecutive keyframes

Solve it with Gauss-Newton with a zero bias seed

2.Scale and Gravity Approximation (excl. accelerometer bias)

Preintegrate velocities and positions, rotating correctly the acceleration measurements compensating the gyroscope bias

consider 3 consecutive keyframes

Solve by SVD

3(N – 2) equations and 4 unknowns, therefore we need at least 4 keyframes

3.Accelerometer Bias Estimation, Scale and Gravity Direction Refinement

G: gravity magnitude, g_I: gravity direction in inertial reference frame I={0, 0, -1}, g

4.Velocity Estimation

substitute the first-order approximation into (2), we get … (lambda 和(2) 一样)

consider 3 consecutive keyframes

solved by SVD

3(N – 2) equations and 6 unknowns and we need again at least 4 keyframes

We could relinearize (first-order approximation) and iterate the solution, but in practice we found that a second iteration does not produce a noticeable improvement.

Bias Reinitialization after Relocalization

ORBSLAM-VI: Experiment & Result

- Dataset: MAV EuRoC 11 seqs (stereo 20Hz IMU 200Hz) scenes: 2 rooms, 1 industrial env; classify to easy/medium/difficult based on illumination, texture, fast/slow motions or motion blur
- Equipment: Intel Core i7-4700MQ, 8GB RAM

	Visual-Inertial ORB-SLAM						Monocular ORB-SLAM	
	No Full BA			Full BA			No Full BA	Full BA
	RMSE (m)	Scale Error (%)	RMSE(m) *GT scale**	RMSE (m)	Scale Error (%)	RMSE (m) *GT scale**	RMSE(m) *GT scale**	RMSE(m) *GT scale**
V1_01_easy	0.027	0.9	0.019	0.023	0.8	0.016	0.015	0.015
V1_02_medium	0.028	0.8	0.024	0.027	1.0	0.019	0.020	0.020
V1_03_difficult	X	X	X	X	X	X	X	X
V2_01_easy	0.032	0.2	0.031	0.018	0.2	0.017	0.021	0.015
V2_02_medium	0.041	1.4	0.026	0.024	0.8	0.017	0.018	0.017
V2_03_difficult	0.074	0.7	0.073	0.047	0.6	0.045	X	X
MH_01_easy	0.075	0.5	0.072	0.068	0.3	0.068	0.071	0.070
MH_02_easy	0.084	0.8	0.078	0.073	0.4	0.072	0.067	0.066
MH_03_medium	0.087	1.5	0.067	0.071	0.1	0.071	0.071	0.071
MH_04_difficult	0.217	3.4	0.081	0.087	0.9	0.066	0.082	0.081
MH_05_difficult	0.082	0.5	0.077	0.060	0.2	0.060	0.060	0.060

**GT scale*: the estimated trajectory is scaled so that it perfectly matches the scale of the ground-truth. These columns are included for comparison purposes but do not represent the output of a real system, but the output of an *ideal* system that could estimate the true scale.

Given: a set of keyframes processed by a Mono SLAM

Require:

- mono SLAM running for a few seconds, sensor performs a motion(有运动, make all variable observable)
- any two consecutive keyframes close in time (to reduce IMU noise integration)

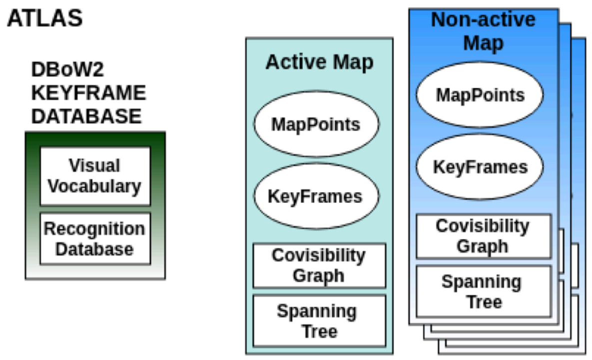
Method to compute an initial estimation for a VI full BA of the scale, gravity direction, velocity and IMU biases

1.Gyroscope Bias Estimation
2.Scale and Gravity Approximation (excl. accelerometer bias)
3.Accelerometer Bias Estimation, Scale and Gravity Direction Refinement
4.Velocity Estimation

Bias Reinitialization after Relocalization

ORBSLAM-Atlas: Overview

- A system able to handle an unlimited number of disconnected sub-maps
  - Preliminaries
    - Multiple-map representation → Altas (it has a unique DBoWs database of keyframes for all sub-maps)
    - sub-maps → Map (active / non-active)



- Operations: new map creation, relocalization in multiple maps, map merging
- Declare camera lost: number of tracked points (previous approaches) + discard inaccurately estimated camera poses due to bad geometrical conditioning (our new)

A system able to handle an unlimited number of disconnected sub-maps

Preliminaries:
- Multiple-map representation → Altas (it has a unique DBoWs database of keyframes for all sub-maps)
    contains a virtually unlimited number of maps,
    a unique for all the maps DBoW2 recognition database (stores all the info to recognize any keyframe in any maps)
- sub-maps → Map (active / non-active)
    sub-maps = keyframes, map points, covisibility graph, spanning tree

Operations: new map creation, relocalization in multiple maps, map merging

Declare camera lost: number of tracked points (previous approaches) + discard inaccurately estimated camera poses due to bad geometrical conditioning (our new)

ORBSLAM-Atlas

- Place recognition stage to detect common map regions
  - Both in active map → loop closure
  - two in different maps → map merge
- Criteria to determine a camera on track (i.e. not lost)
  - 1. number of matches features above a threshold (between current frame and points in local map)
  - 2. Camera pose observability (estimated by camera pose error covariance)

$$\begin{aligned}\mathbf{T}_{i,w} &= \mathrm{Exp}\left(\boldsymbol{\varepsilon}_i\right) \oplus \hat{\mathbf{T}}_{i,w} \\ \boldsymbol{\varepsilon}_i &= \begin{pmatrix} x & y & z & \omega_x & \omega_y & \omega_z \end{pmatrix} \sim \mathcal{N}(0, \mathbf{C}_i) \\ \mathbf{H}_i &\simeq \sum_{j=1}^{m_i} \mathbf{J}_{i,j}^{\intercal} \boldsymbol{\Omega}_{i,j} \mathbf{J}_{i,j} \\ \mathbf{C}_i &= \mathbf{H}_i^{-1}\end{aligned}$$

$$\begin{aligned}\max\left(\sigma_x, \sigma_y, \sigma_z\right) &< \sigma_{th}^t \\ \left[\begin{array}{cccccc}\sigma_x^2 & \sigma_y^2 & \sigma_z^2 & \sigma_{\omega_x}^2 & \sigma_{\omega_y}^2 & \sigma_{\omega_z}^2\end{array}\right] &= \mathrm{diag}\left(\mathbf{C}_i\right)\end{aligned}$$

- New map creation
  - Camera tracking lost → relocalize in Atlas → if relocalization unsuccessful for a few frames → active map becomes non-active (still stored in Atlas) → new map initialization

Camera Pose Observability

Assume map points perfectly estimated (reduce complexity and facilitate real-time operation)

measurement information matrix Ω: uncertainty for the observation x(i, j) of the map points j in camera I
T_tilt: estimated camera pose 6DOF (SE(3))
ε_i: unbiased Gaussian vector (defines Lie algebra approximating T around T_tilt)
Exp: parameters space to Lie group
C: covariance matrix (camera estimation accuracy)
J: Jacobian matrix for the camera pose measurement due to the observation of the map point j in the camera I

As translation is the weakly observable magnitude, use only diagonal values of C corresponding to the translation error.

ORBSLAM-Atlas

- Relocalization in multiple maps
  - Query Atlas DBoW database → similar keyframe (candidate) in a map → estimate camera pose (PnP + RANSAC + search for matches + non-linear camera pose only optimization)
- Seamless Map Merge (active map swallows the other, merged map replaces the 2 merging maps)
  In parallel to tracking, LM, (occasionally) global BA thread; Before merging, LM & global BA stopped
  - Step 1: Detection of common area between 2 maps
    Place recognition:
    - Repeatedly for 3 keyframes connected by the covisibility graph (reduce FP)
    - Provides 2 matching keyframes and set of matched points in 2 maps
  - Step 2: Estimation of aligning transformation between world reference frames of two merging map T_{as}, (Sim(3)/SE(3))
    Initial estimation (by map points) → Match points of M_a in K_s → Final estimation by NL optimization
  - Step 3: Combining the merging maps
    Apply T_{as} to K_s & M_s → fuse duplicate map points → combine all keyframes & mappoints in a & s to m → merge spanning tree and covisibility graph
  - Step 4: Local BA the welding area
    Fixed keyframes that were fixed in map a, optimize the rest
    Another duplicate mappoints fusion, update the merged map m covisibility graph
  - Step 5: Pose graph optimization of the merged map m

Relocalization

Seamless Map Merge (active map swallows the other, merged map replaces the 2 merging maps)
- LM & global BA stopped
  - reason: LM (avoid new keyframes insertion in Atlas), global BA (spanning tree might change)

ORBSLAM-Atlas: Experiment & Result

- Quantitative Evaluation on EuRoC (RMS ATE)

	ORBSLAM-Atlas Monocular			ORBSLAM Monocular			ORBSLAM-Atlas Stereo			ORBSLAM2 Stereo		
	ATE (m)	Cover (%)	# Maps	ATE (m)	Cover (%)	# Maps	ATE (m)	Cover (%)	# Maps	ATE (m)	Cover (%)	# Maps
V1_03	**0.106**	**90.74**	2	0.132	10.32	1	0.051	100	1	**0.046**	100	1
V2_03	**0.093**	**70.74**	2	0.146	15.71	1	**0.218**	**94.55**	5	0.316	89.21	1

TABLE I: Performance on the difficult Vicon Room EuRoC datasets. RMS ATE in meters. Median values after 5 runs.

	ORBSLAM-Atlas stereo	VINS stereo	VINS Mono Inertial
V1_01	**0.036**	0.550	0.068
V1_02	**0.022**	0.230	0.084
V1_03	**0.051**	X	0.190
V2_01	**0.034**	0.230	0.081
V2_02	**0.028**	0.200	0.150
V2_03	**0.218**	X	0.220
MH_01	**0.036**	0.540	0.120
MH_02	**0.021**	0.460	0.120
MH_03	**0.026**	0.330	0.130
MH_04	**0.103**	0.780	0.180
MH_05	**0.054**	0.500	0.210
muliple-session MH_01-MH_05	**0.086**	-	0.210

TABLE II: Multiple-session performance on EuRoC datasets. We report the results of the individual mapping sessions, and the global multi-session map after the sequential processing of datasets MH_01 to MH_05. Reported RMS ATE (m) are median values after 5 runs.

	Global map RMS ATE (m)
CCM-SLAM (Mono*)	0.077
ORBSLAM-Atlas (Mono*)	**0.024**
ORBSLAM-Atlas (Stereo)	0.035

TABLE III: RMS ATE (m) in the EuRoC Machine Hall (MH_01, MH_02 and MH_03). * indicates that the aligning transformation prior to ATE computation includes a scale correction. The reported values are the average after 5 runs to make them comparable with results reported in [14].

TODO!!!

IMU-Initialization

-