Visual-Inertial Navigation

Graduate Course INTR-6000P

Week 8 - Lecture 16

Changhao Chen

Assistant Professor

HKUST (GZ)

Motivation: Why Fuse Vision and IMU?

Cameras: accurate geometric constraints, but fail under:
•Motion blur
•Low texture
•Low light

IMUs: high-frequency motion tracking, but suffer from **drift**.

Combined: complementary strengths → robust, consistent motion estimation.

Source: Visual-Inertial Odometry of Aerial Robots

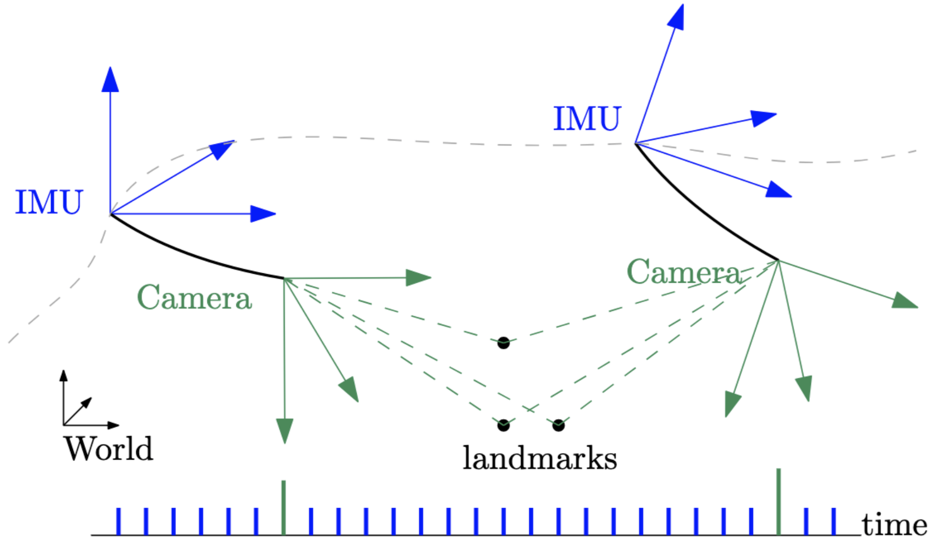
Visual–Inertial Navigation

Fuse high-rate IMU propagation with low-rate visual corrections.

Fusion frameworks:
•**Filter-based** (MSCKF, ROVIO)
•**Optimization-based** (OKVIS, VINS-Mono)
Each provides **pose, velocity, and bias estimation**.

The Role of IMU in Vision-Based Systems:

•Stabilizes fast motion when frames are blurred.
•Provides initial prediction for visual tracking and optimization.
•Enhances scale observability in monocular setups.



Visual–Inertial Navigation

Pipeline:

Image acquisition → feature extraction/tracking
IMU propagation → predict next state
Visual update → correct drift
Optimization or filtering → refine pose

System State Definition:

$$\mathbf{x} = [\mathbf{R}, \mathbf{v}, \mathbf{p}, b_g, b_a, \text{landmarks}]$$

- Orientation $\mathbf{R}$, velocity $\mathbf{v}$, position $\mathbf{p}$.
- IMU biases b_g, b_a.
- Optional: feature landmarks for visual constraints.

IMU Measurement Models

- **Gyroscope:**

$$\boldsymbol{\omega}_m = \boldsymbol{\omega}_{WB} + \mathbf{b}_g + \mathbf{n}_g$$

  - $\boldsymbol{\omega}_m$: measured angular velocity
  - $\boldsymbol{\omega}_{WB}$: true angular velocity (world → body)
  - $\mathbf{b}_g$: gyro bias (slowly varying)
  - $\mathbf{n}_g$: white Gaussian noise
- **Accelerometer:**

$$\mathbf{f}_m = \mathbf{R}_W^B(\mathbf{a}_W - \mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a$$

  - $\mathbf{f}_m$: measured specific force
  - $\mathbf{a}_W$: true linear acceleration of IMU
  - $\mathbf{b}_a, \mathbf{n}_a$: bias and noise terms

Visual Measurement Model

$$\mathbf{z}_{ij} = \begin{bmatrix} u \\ v \end{bmatrix}_{ij} = \pi\left(\mathbf{K}\,\mathbf{R}_W^{C_i}(\mathbf{p}_W^j - \mathbf{p}_W^{C_i})\right) + \mathbf{n}_v$$

where

- $\mathbf{z}_{ij}$: image projection of landmark j in camera i
- $\pi(\cdot)$: perspective projection function
- $\mathbf{K}$: camera intrinsics
- $\mathbf{R}_W^{C_i}, \mathbf{p}_W^{C_i}$: camera pose in world frame
- $\mathbf{p}_W^j$: 3D landmark position
- $\mathbf{n}_v$: measurement noise

Combined Measurement Fusion

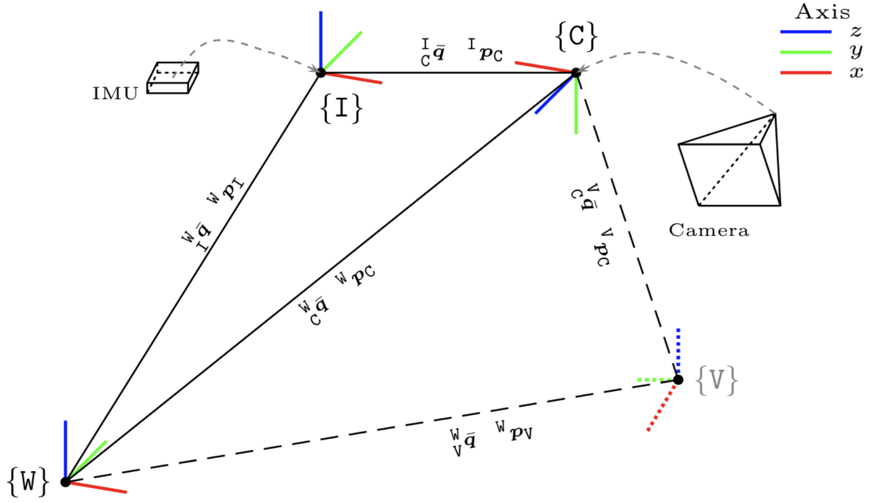
Both sensor types form constraints on the state:

$$\mathbf{x}_k = \{\mathbf{R}_W^{B_k}, \mathbf{p}_W^{B_k}, \mathbf{v}_W^{B_k}, \mathbf{b}_{g_k}, \mathbf{b}_{a_k}\}$$

- IMU: provides **motion continuity** between states ($k \to k+1$)
- Camera: provides **observation constraints** to landmarks or previous frames

$$\mathbf{u} = K[R_{CW}|t_{CW}]\mathbf{P}_W$$

Relates 3D landmark position to 2D image coordinates.
Nonlinear constraint linking vision and motion.



Source: A review of visual inertial odometry from filtering and optimisation perspectives

IMU Propagation (Prediction Step)

IMU data used to **propagate** state forward.
Apply preintegration:

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1}, \mathbf{u}_{IMU})$$

Result: predicted orientation, velocity, and position before camera correction.

Problem: Reintegrating IMU data for each optimization iteration is expensive.

Solution: *IMU preintegration* compresses all IMU data between keyframes.

IMU Preintegration

Preintegrated Quantities

$$\Delta\mathbf{R}_{ij},\ \Delta\mathbf{v}_{ij},\ \Delta\mathbf{p}_{ij}$$

Accumulate small motion increments:

$$\Delta\mathbf{R}_{ij} = \prod_{k=i}^{j-1} \exp([\omega_k - b_g]\Delta t)$$

Preintegration Equations

$$\begin{aligned}
\Delta\mathbf{R}_{ij} &= \mathbf{R}_i^T\mathbf{R}_j \\
\Delta\mathbf{v}_{ij} &= \mathbf{R}_i^T(\mathbf{v}_j - \mathbf{v}_i - g\Delta t_{ij}) \\
\Delta\mathbf{p}_{ij} &= \mathbf{R}_i^T(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}g\Delta t_{ij}^2)
\end{aligned}$$

Preintegration Jacobians and Covariance

$$\mathbf{z}_{IMU} = f(\mathbf{x}_i, \mathbf{x}_j, b_a, b_g) + n$$

Linearize preintegrated terms w.r.t. bias.
Maintain covariance of preintegration errors.
Enables use in optimization as a single factor.

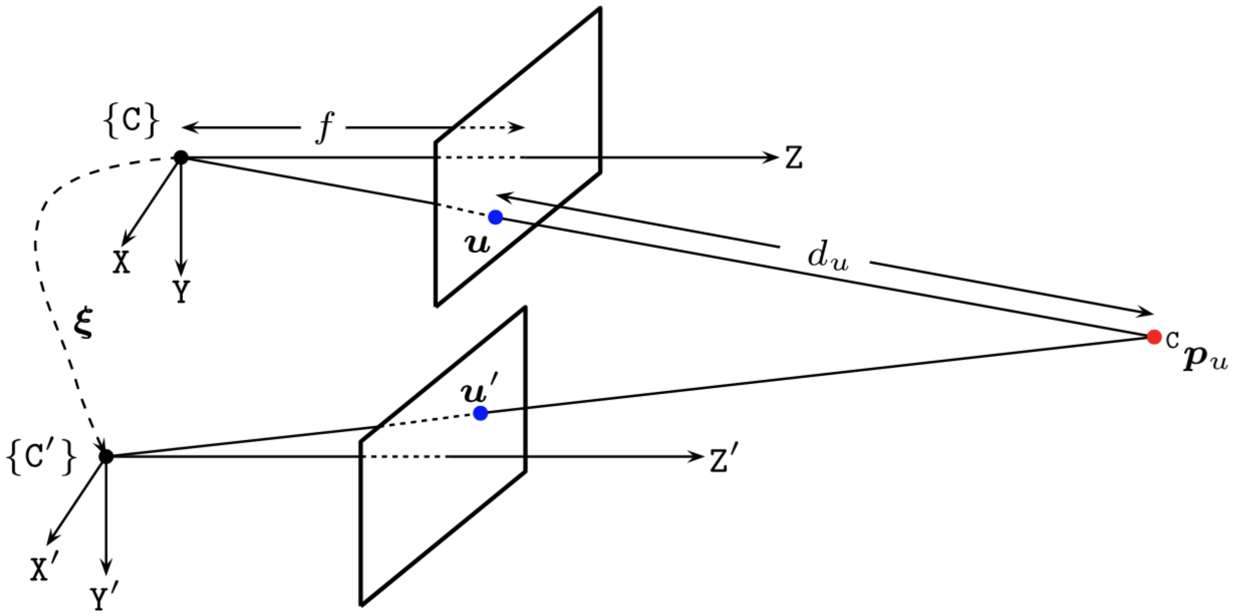
Vision Update (Correction Step)

Visual features used to correct accumulated IMU error.

For each observed feature:

$$\mathbf{r}_{cam} = \mathbf{z}_{obs} - \pi(\mathbf{R}, \mathbf{p}, \mathbf{P})$$

Apply residual minimization to refine state.



Fusion Frameworks

1) **Filtering-based:**
sequential estimation (EKF-style).

1) **Optimization-based:**
nonlinear least-squares across time window.

Trade-offs: computational cost vs. accuracy.

Loosely Coupled Fusion:

•Fuse separate **visual odometry** (VO) and **IMU** estimates.

•VO provides relative motion or pose; IMU provides short-term propagation.

•Use EKF or complementary filter for fusion.

Pros: simple, modular.

Cons: less accurate, partial observability.

Tightly Coupled Fusion

•Jointly optimizes IMU and visual feature measurements.

•Single estimation problem with shared state.

•Adds both **visual reprojection** and **IMU preintegration** residuals.

Pros: high accuracy and consistency.

Cons: higher computation.

Filtering-based Method

MSCKF - Multi-State Constraint Kalman Filter (2007)

Principle:

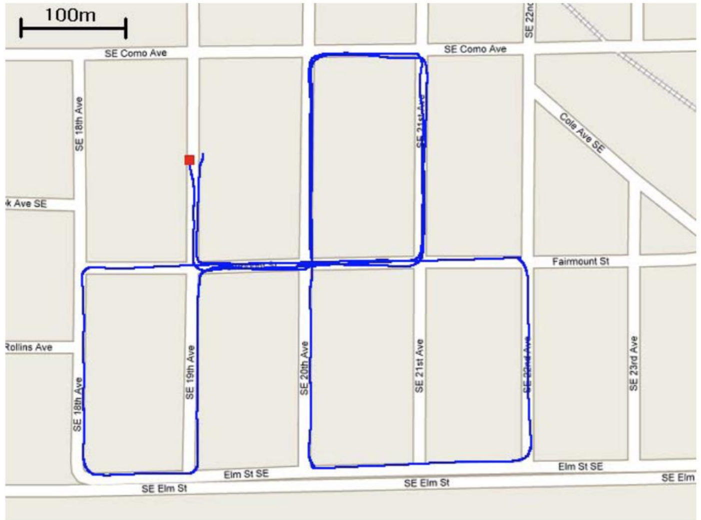
•Predict step: propagate IMU states.

•Update step: correct with visual feature reprojection errors.

State augmentation: camera poses added temporarily.

Pros: real-time, low latency.

Cons: linearization errors accumulate; harder to relinearize.



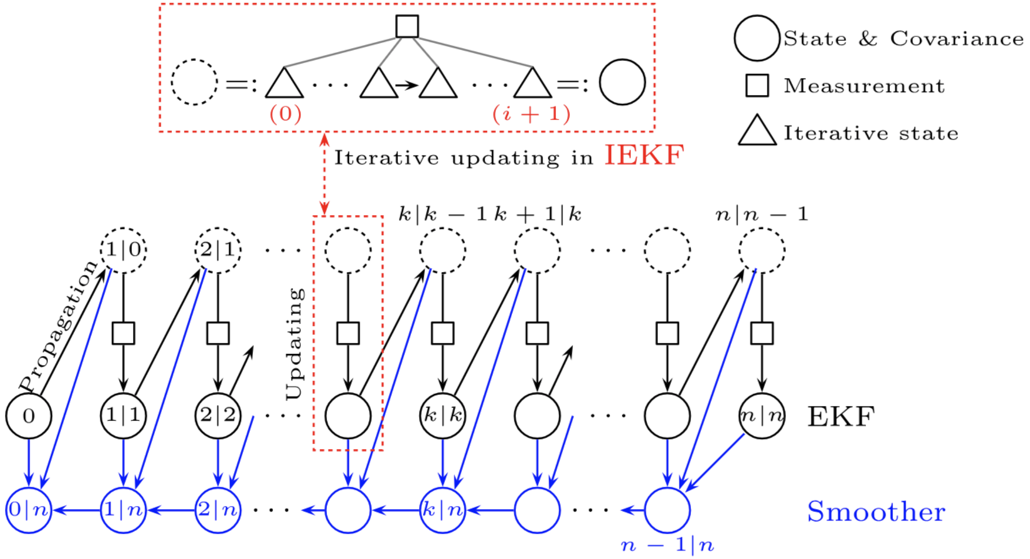
Algorithm 1 Multi-State Constraint Filter

Propagation: For each IMU measurement received, propagate the filter state and covariance (cf. Section III-B).

Image registration: Every time a new image is recorded,

- augment the state and covariance matrix with a copy of the current camera pose estimate (cf. Section III-C).
- image processing module begins operation.

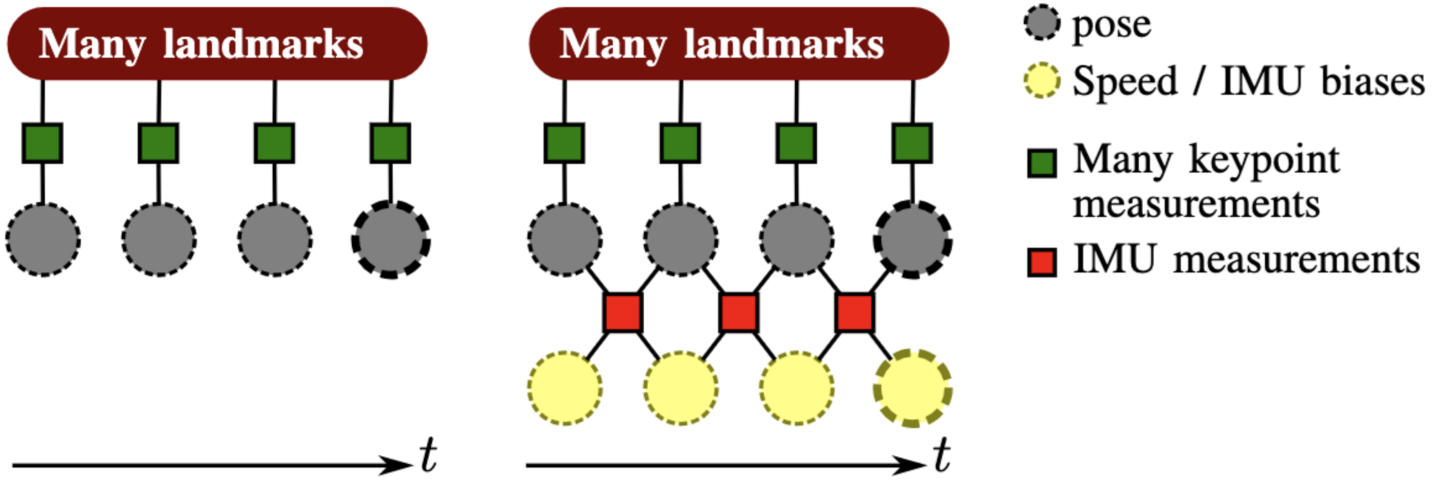
Update: When the feature measurements of a given image become available, perform an EKF update (cf. Sections III-D and III-E).



Optimization based Method

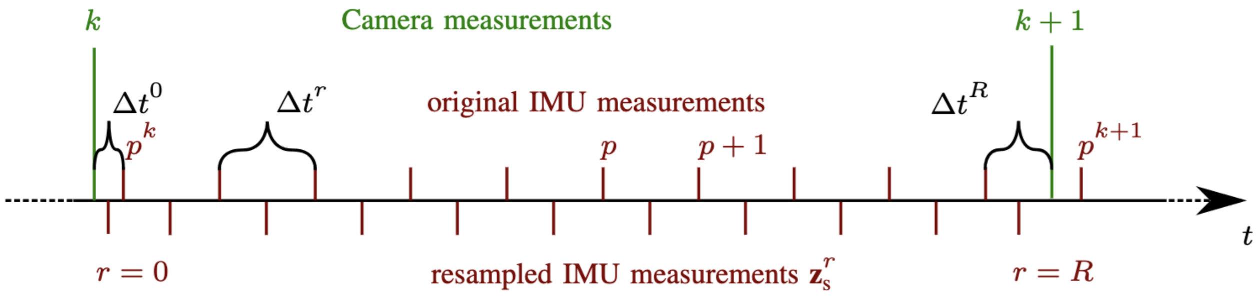
OKVIS - Keyframe-based visual–inertial odometry using nonlinear optimization (2015)

- Optimization-based keyframe VIO.
- Nonlinear least-squares over window of keyframes.
- Uses preintegrated IMU measurements between keyframes.



Maintain window of recent states (e.g., 10 keyframes).
Minimize cost function:

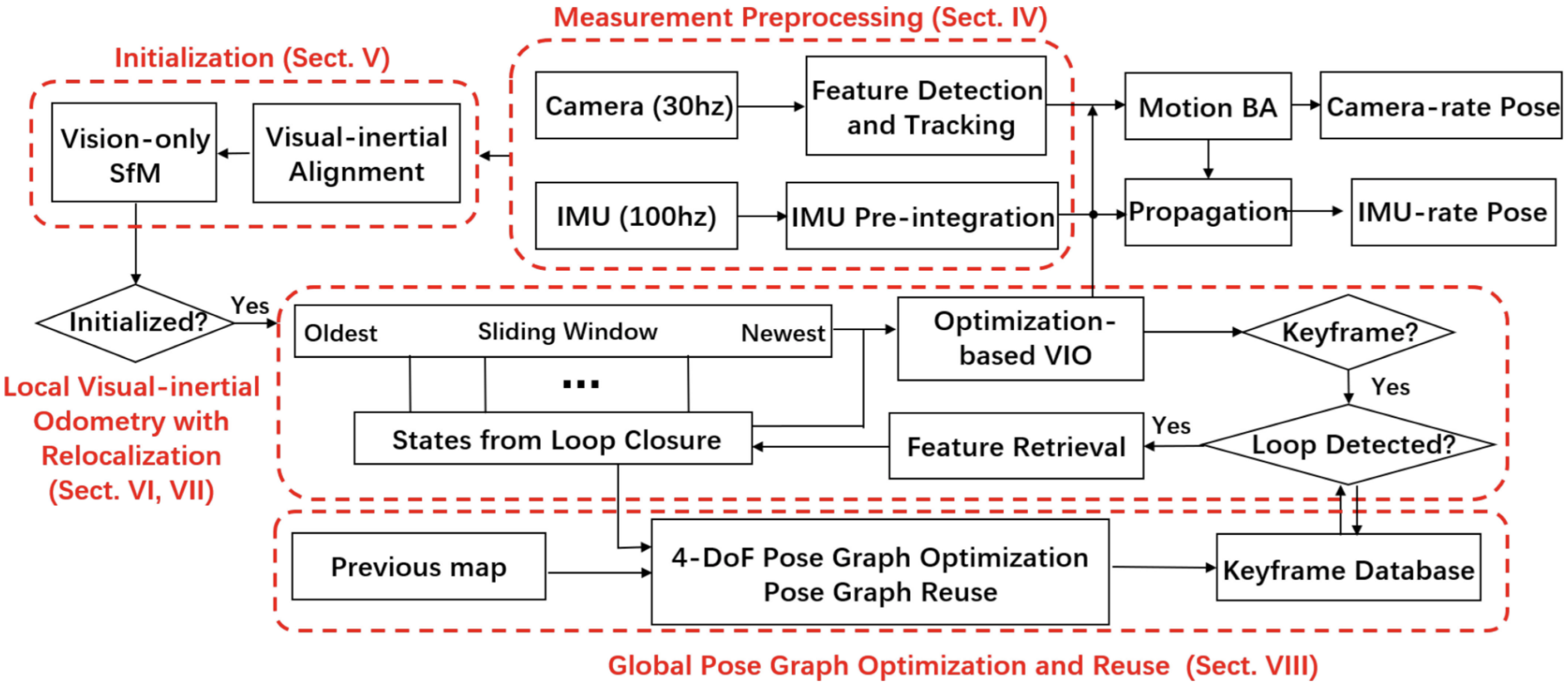
$$J = \sum \|\mathbf{r}_{ij}^{v}\|^2 + \sum \|\mathbf{r}_{k,k+1}^{imu}\|^2$$

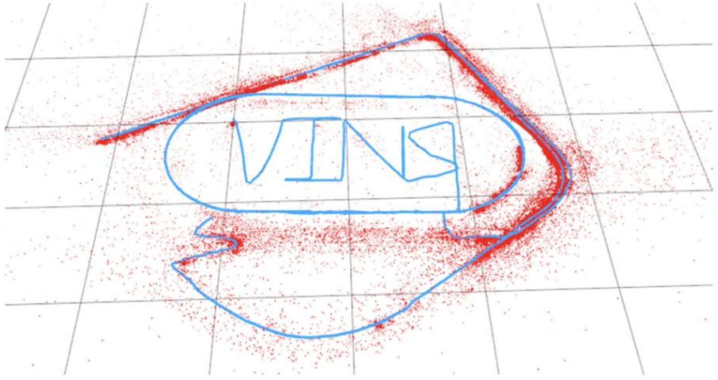


Optimization based Method

VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator (2018)

Combines:
•IMU preintegration (Forster et al.)
•Sliding window optimization
•Loop closure and global pose graph





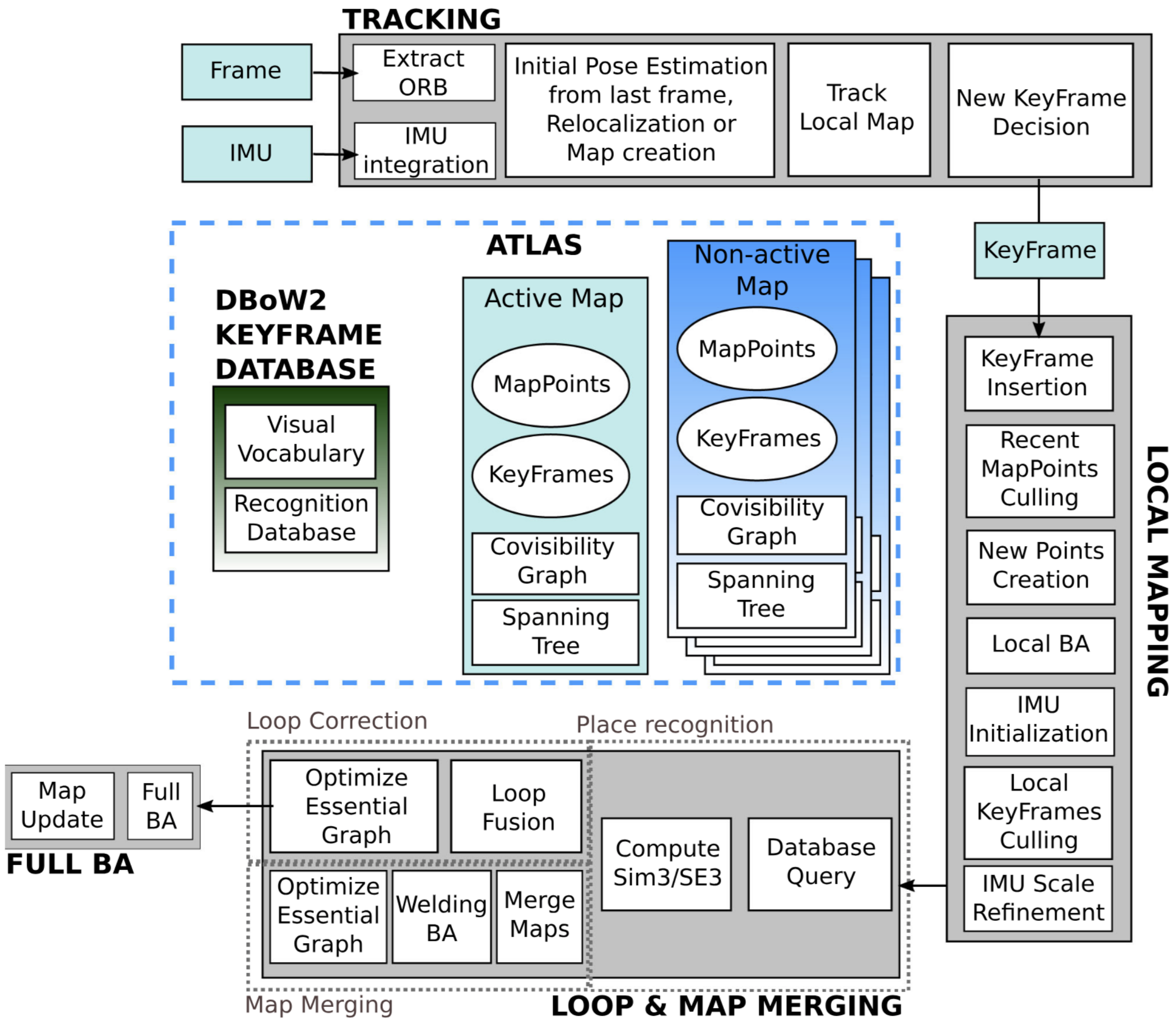
(a)


(b)

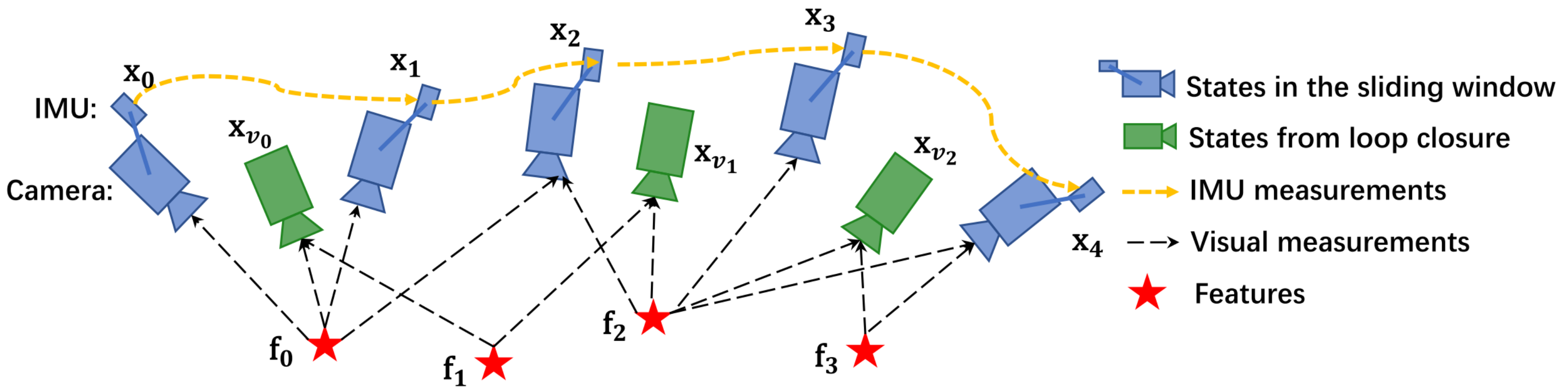
Optimization based Method

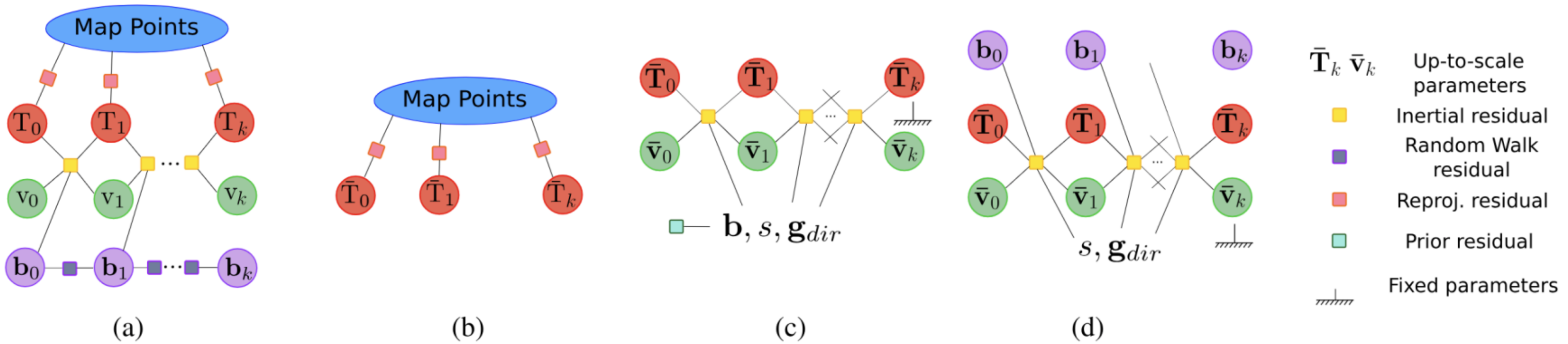
ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM (2021)

- Unified visual, visual–inertial, and multi-map SLAM.
- Incorporates IMU tightly in graph optimization.
- Robust initialization and real-time loop closure.



Factor Graph Representation





Nodes: camera/IMU poses, biases, landmarks.
Edges:
•Visual constraints (reprojection residuals).
•IMU preintegration constraints.

$$\min_{\mathbf{x}} \sum \|r_{vision}\|^2 + \sum \|r_{IMU}\|^2$$

Initialization: The Critical Bootstrap

We need initial values for all states, landmarks, *and* IMU biases to start the optimization. This is hard, especially for monocular VIO.

•**Standard Procedure:**

- **Pure Visual SLAM:** Run a vision-only SLAM for a few seconds.
- **Align with Gravity:** The vision-only scale is arbitrary. The IMU's gravity vector provides the absolute "down" direction. Align the visual map with gravity.
- **Recover Scale:** Use the known magnitude of gravity (9.81 m/s^2) to recover the metric scale of the visual map.
- **Initialize Velocity and Biases:** Solve a small linear system to get initial velocities and IMU biases.

Challenges

1) Initialization Challenges:

Need to estimate:

- Gravity direction
- Scale (for monocular)
- IMU biases
- Extrinsic calibration

2) Calibration: Camera–IMU Extrinsics:

•Estimate rotation and translation between camera and IMU.

•Must be accurate (errors → drift).

•Use **Kalibr** or similar tools.

3) Time Synchronization

•Camera and IMU timestamps must align (<1 ms offset).

•Time offset causes reprojection errors.

4) Robustness in Real-World Conditions

•**Dynamic scenes:** outlier rejection.

•**Motion blur:** IMU helps maintain tracking.

•**Rolling shutter:** model or compensate during optimization.

•**Failure recovery:** relocalization and loop closure.

Toward Multisensory Fusion

Integration with:

- GNSS → absolute positioning
- LiDAR → depth and map consistency
- Wheel encoders → low-speed drift correction

Discussion:

- How to fuse more sensors to estimate states?
- Why is tightly coupled fusion more accurate?

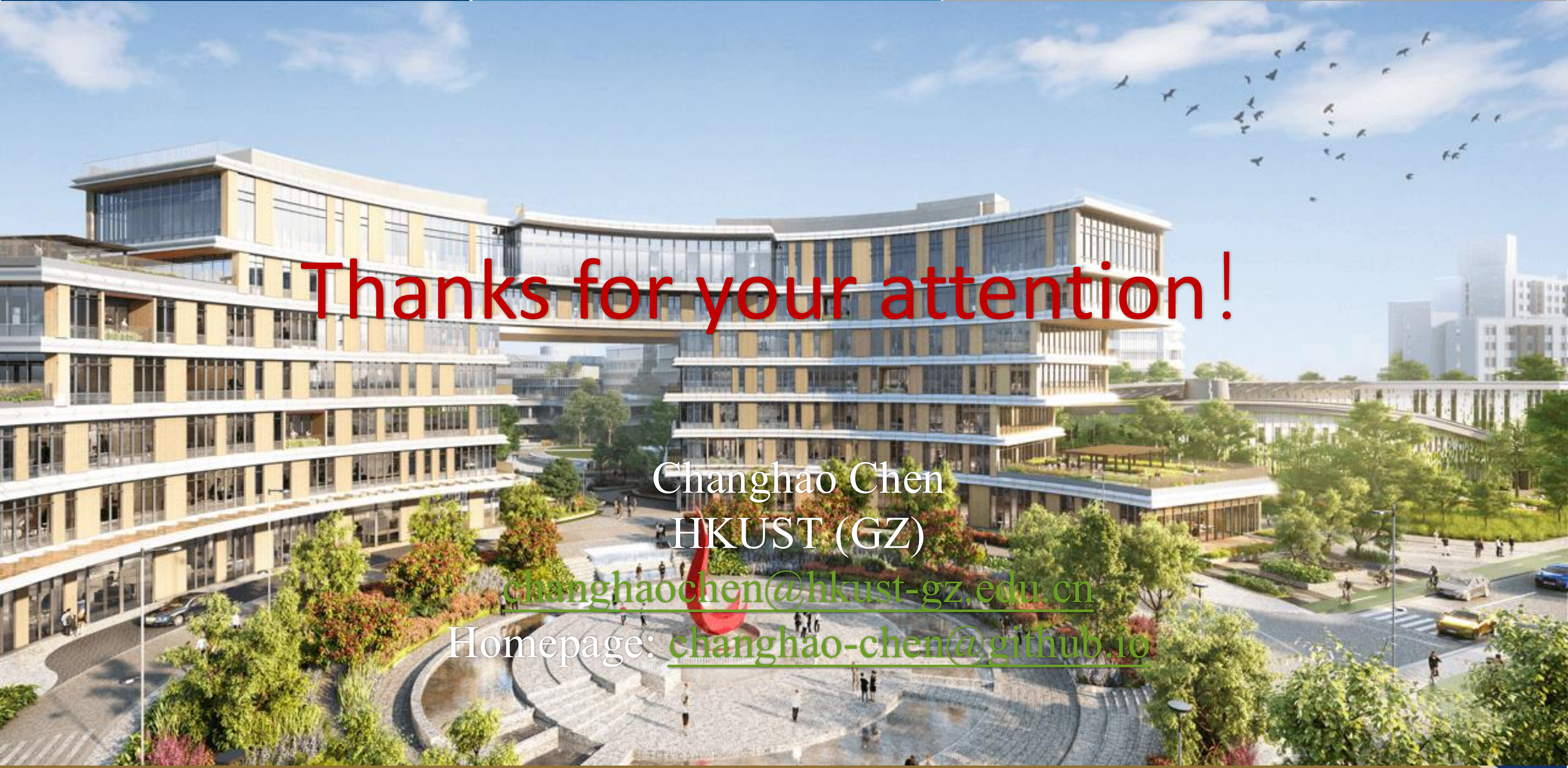
Summary

1) IMU provides short-term prediction; vision corrects long-term drift.

2) Tightly coupled fusion improves accuracy and robustness.

3) Modern VIO systems use preintegration and optimization frameworks.

4) Calibration and initialization are critical for success.



Thanks for your attention！

Changhao Chen
HKUST (GZ)
changhaochen@hkust-gz.edu.cn
Homepage: changhao-chen@github.io