embedded VISION SUMMIT 2018

Visual Inertial Tracking for AR

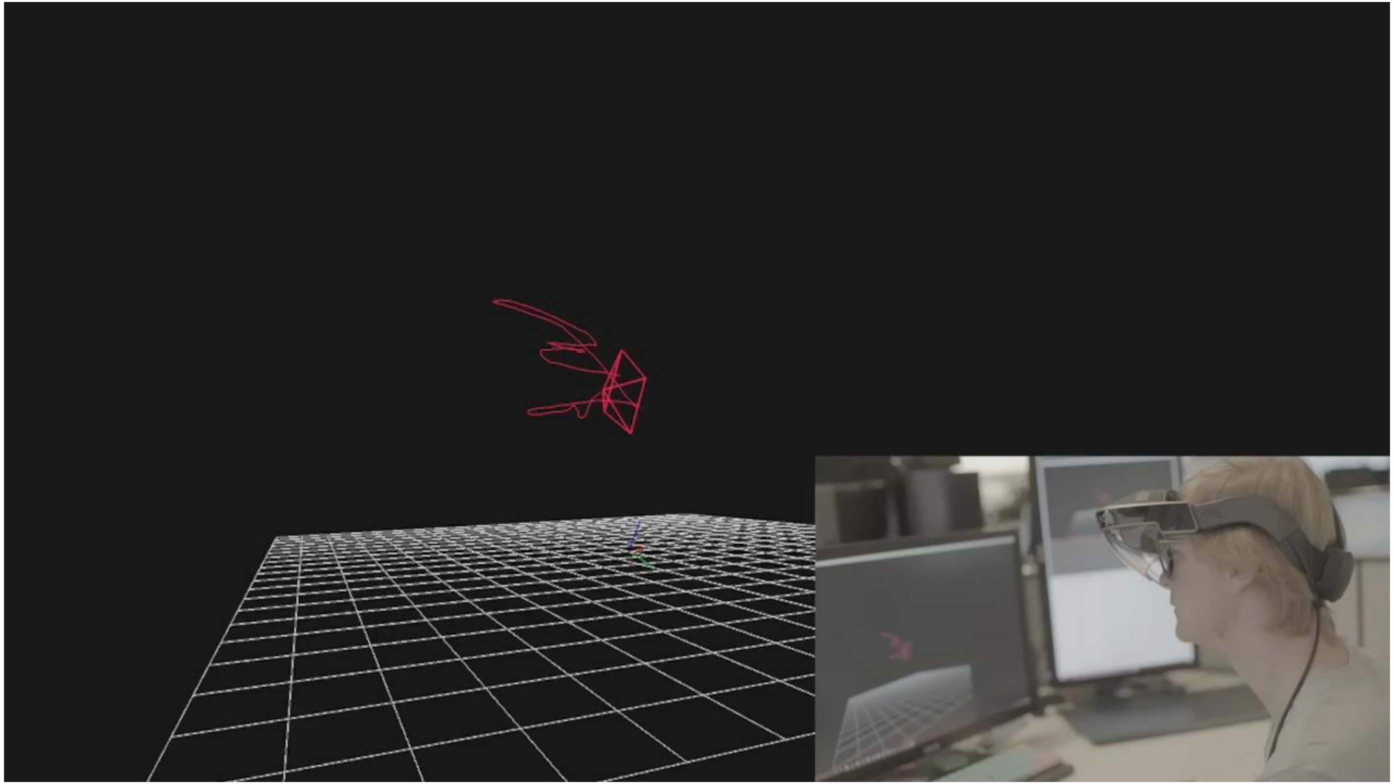
Timo Ahonen, Meta Co.
t.ahonen@metavision.com

May 22, 2018

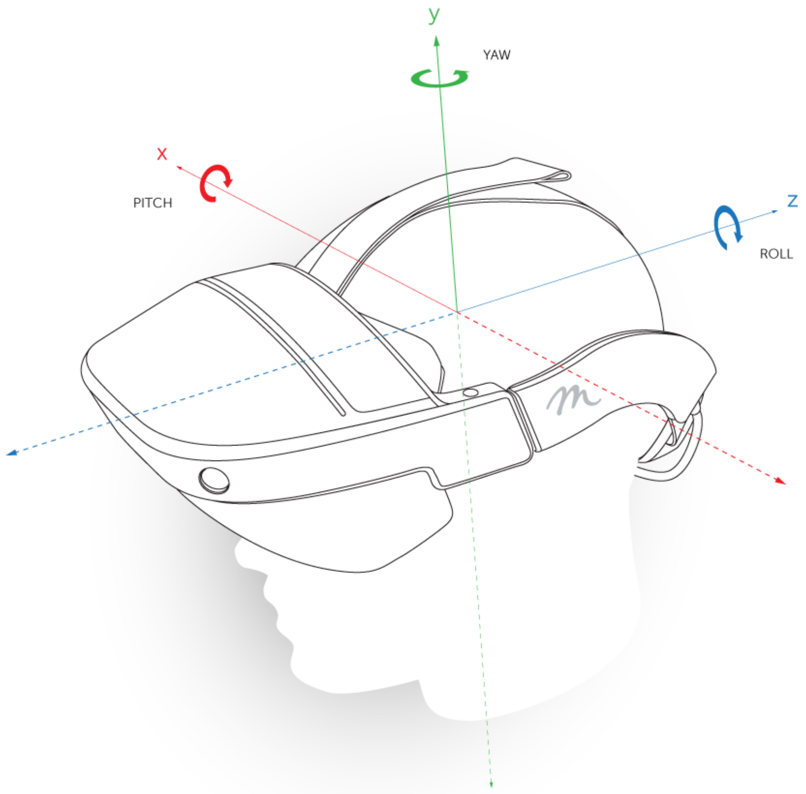
Outline

1. Pose Tracking for Augmented Reality Applications
2. Inertial Sensors and Pose Estimation
3. Introduction to Visual SLAM
4. Visual-Inertial Fusion
5. Applications and Future Challenges

Pose Tracking for Augmented Reality



Pose Tracking: Position and Orientation

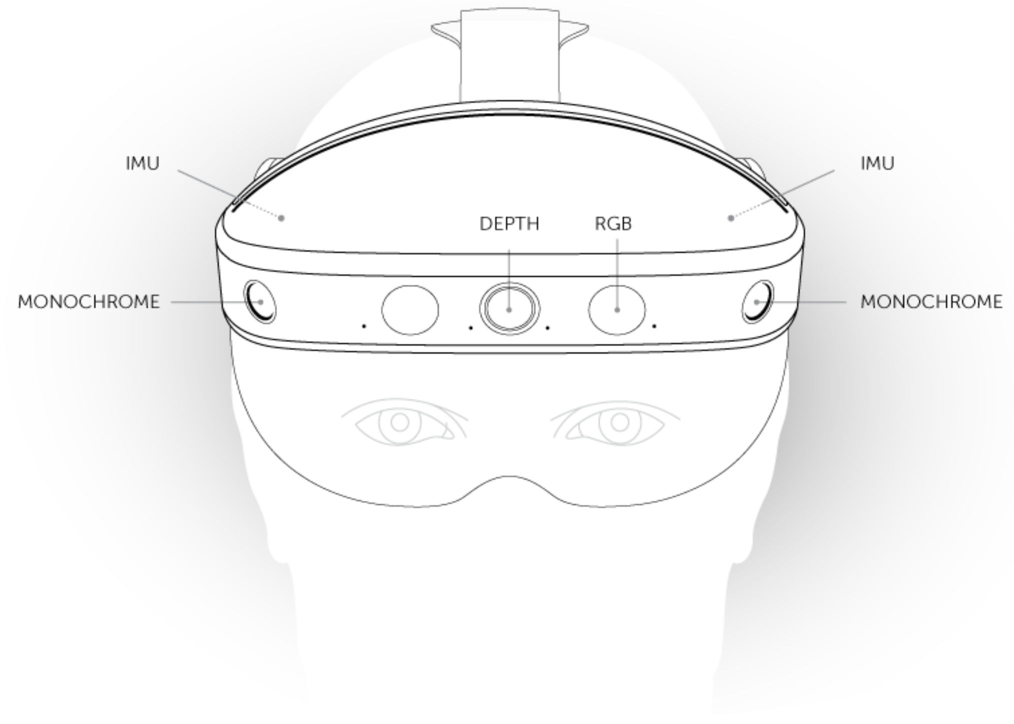


- Goal: track device pose in world coordinates (world frame)

6-degree-of-freedom Headset Pose

- 3 for translation
- 3 for orientation: yaw, pitch, and roll

Modalities for Pose Tracking



- Optical: inside-out or outside-in
- Inertial
- Also: GPS, magnetic, acoustic, …

This presentation: Inside-out visual + inertial tracking

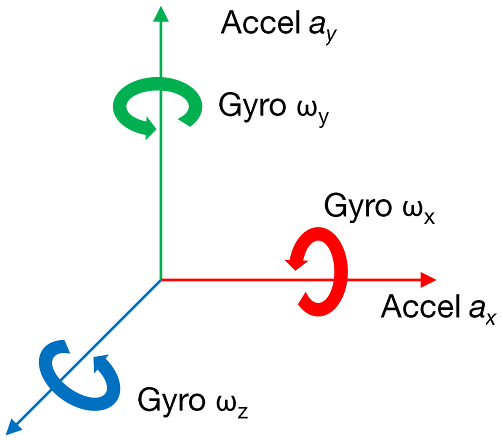
Coordinate Frames
- World frame
- Headset frame
- Sensor frame (each camera, IMU)

Outline

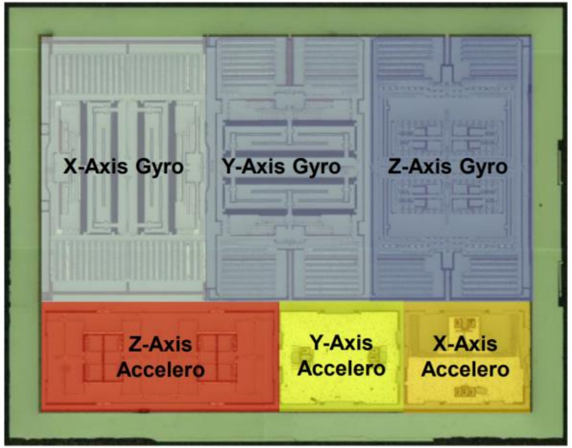
1. Pose Tracking for Augmented Reality Applications
2. Inertial Sensors and Pose Estimation
3. Introduction to Visual SLAM
4. Visual-Inertial Fusion
5. Applications and Future Challenges

Inertial Measurement Unit (IMU)

- Angular Velocity $\boldsymbol{\omega}=(\omega_x, \omega_y, \omega_z)$
- Linear Acceleration $\boldsymbol{a}=(a_x, a_y, a_z)$



MEMS IMUs for smartphones, AR glasses, …



InvenSense MPU-6000 Reverse Costing Report
System Plus Consulting

IMU Measurement Model

Angular Velocity Measurement = True angular velocity

+ Bias

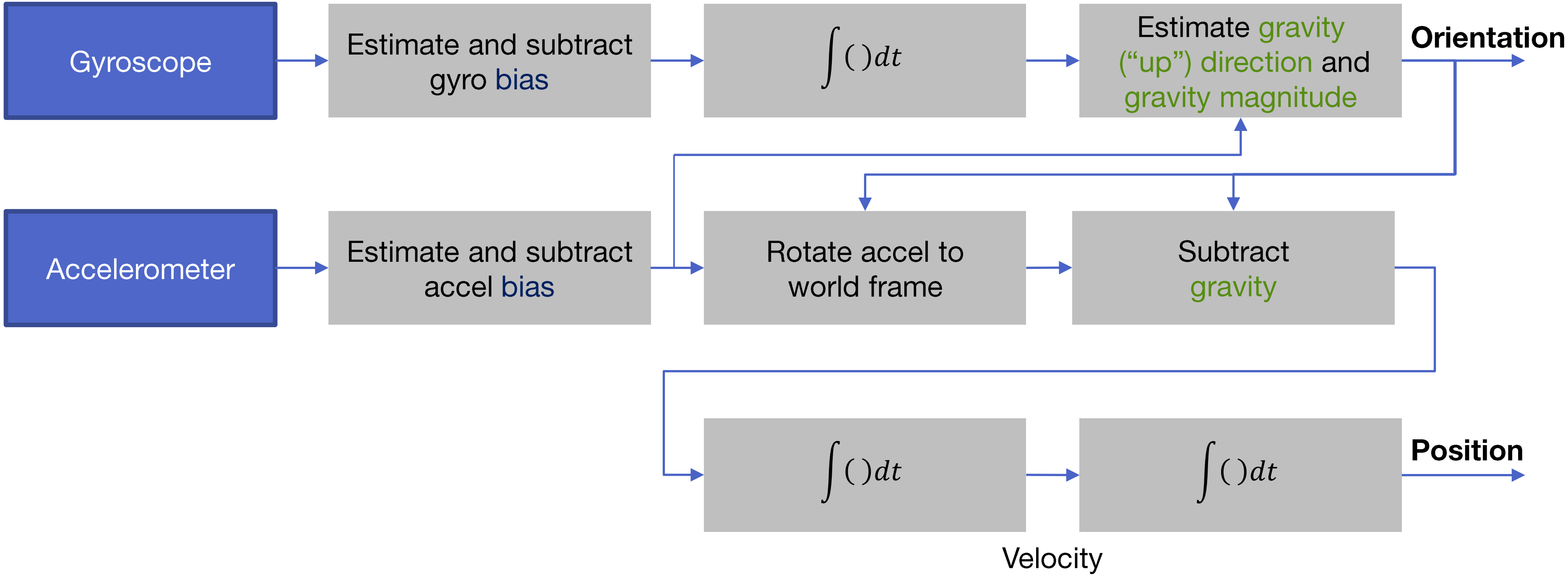
+ Noise

Acceleration Measurement = **Rotate**(True acceleration - Gravity)

+ Bias

+ Noise

Poses from the IMU - Simplified



Poses from the IMU - Notes

- The presented model is simplified. More realistic implementation:
  - System state: orientation, position, velocity, biases, (gravity)
  - State estimation using, e.g., Extended Kalman Filter
- Pros:
  - Low latency
  - High frequency
  - Robust to different motions and environmental conditions
- Cons:
  - Ambiguity between gravity and acceleration
  - Double integration of acceleration leads to large drift – *practically unusable for position tracking*

=> Need another modality to constrain poses – enter visual tracking.

Outline

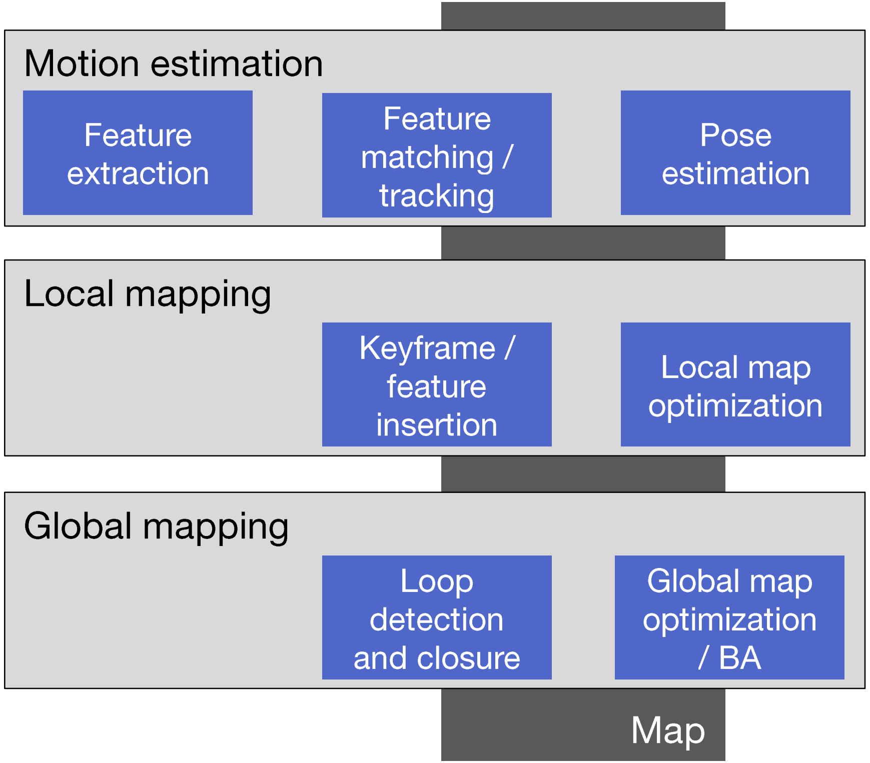
1. Pose Tracking for Augmented Reality Applications
2. Inertial Sensors and Pose Estimation
3. Introduction to Visual SLAM
4. Visual-Inertial Fusion
5. Applications and Future Challenges

Visual SLAM

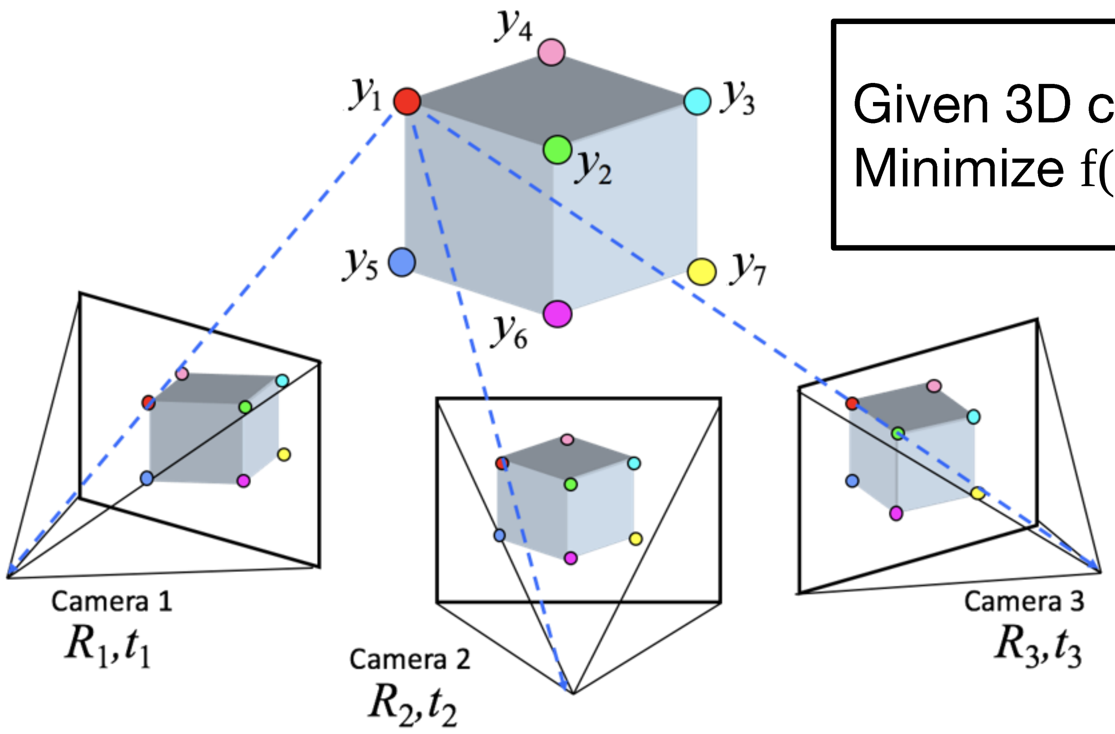
- Chicken-and-egg problem:
  - Estimate camera (headset) pose based on the environment
  - Estimate environment based on the pose
- Temporal component (tracking)

- Direct / indirect
- Dense / sparse
- Filtering / optimization

Real-time Visual SLAM System Architecture

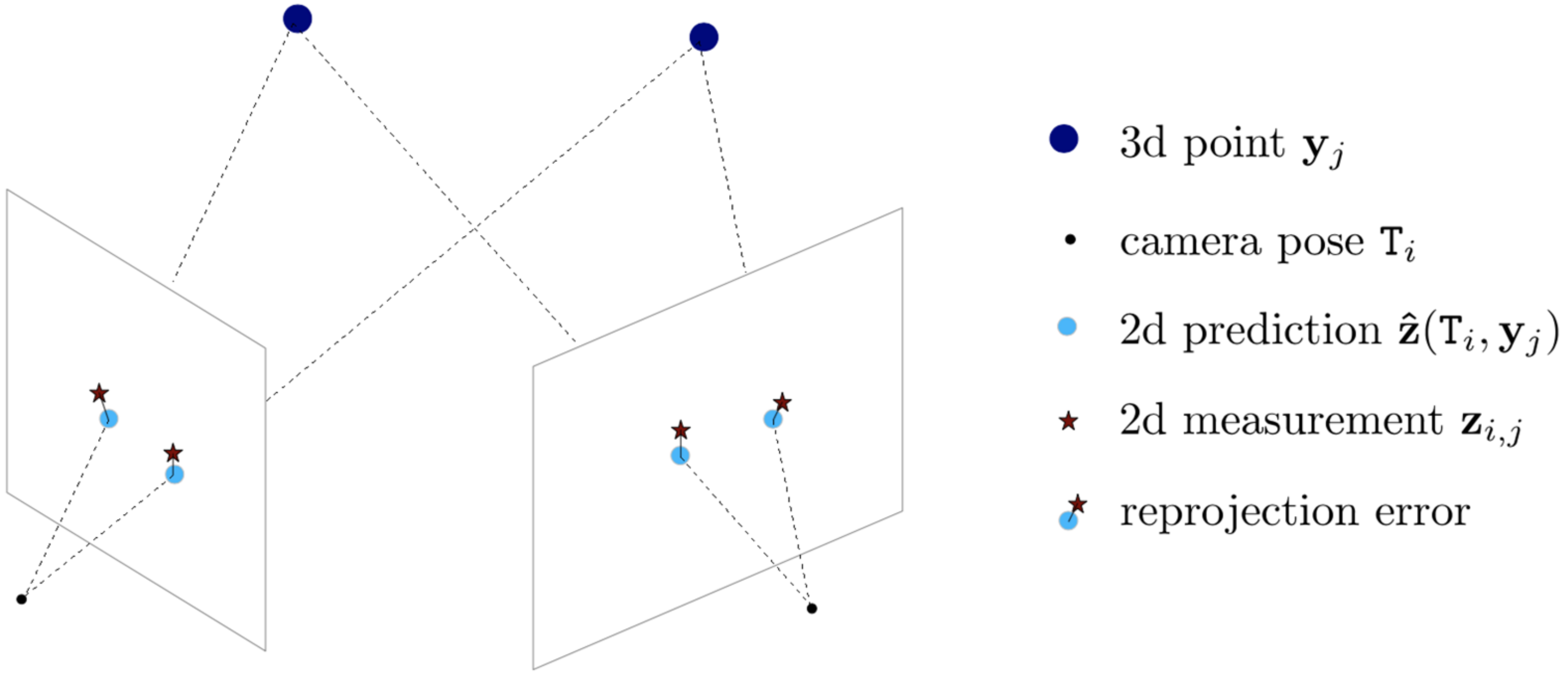


Pose Estimation (Motion-only Bundle Adjustment)



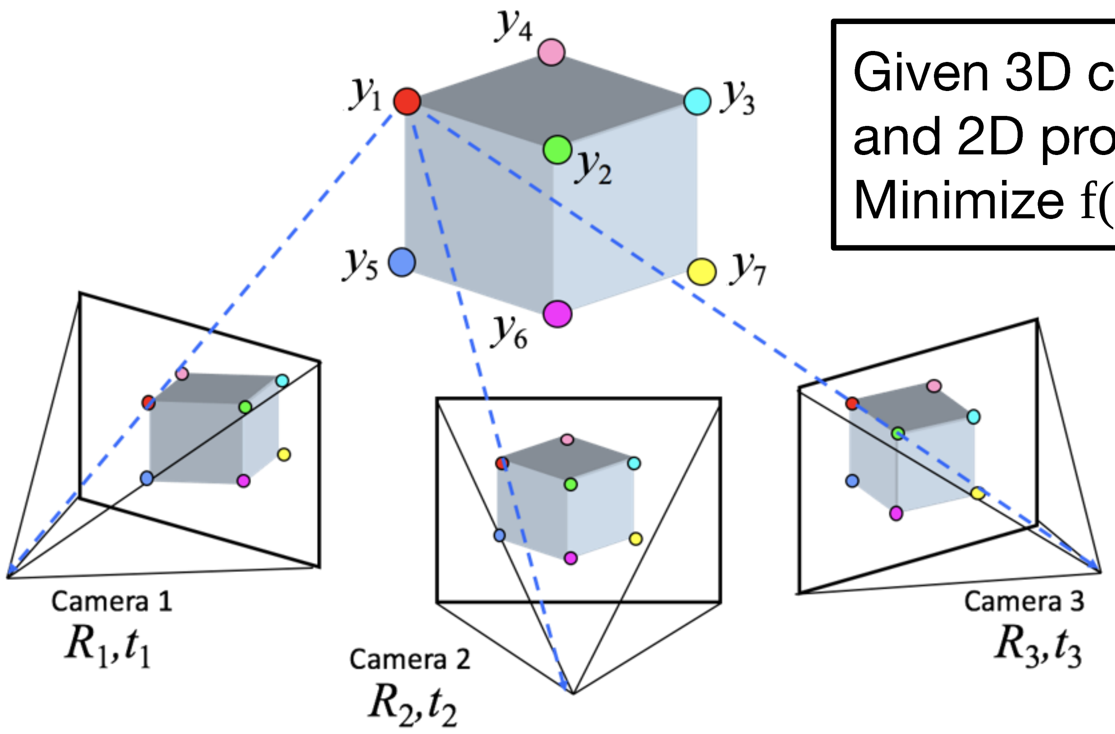
Given 3D coordinates $\mathbf{Y}$
Minimize $f(\mathbf{R}, \mathbf{T}, \mathbf{Y})$

Reprojection Error



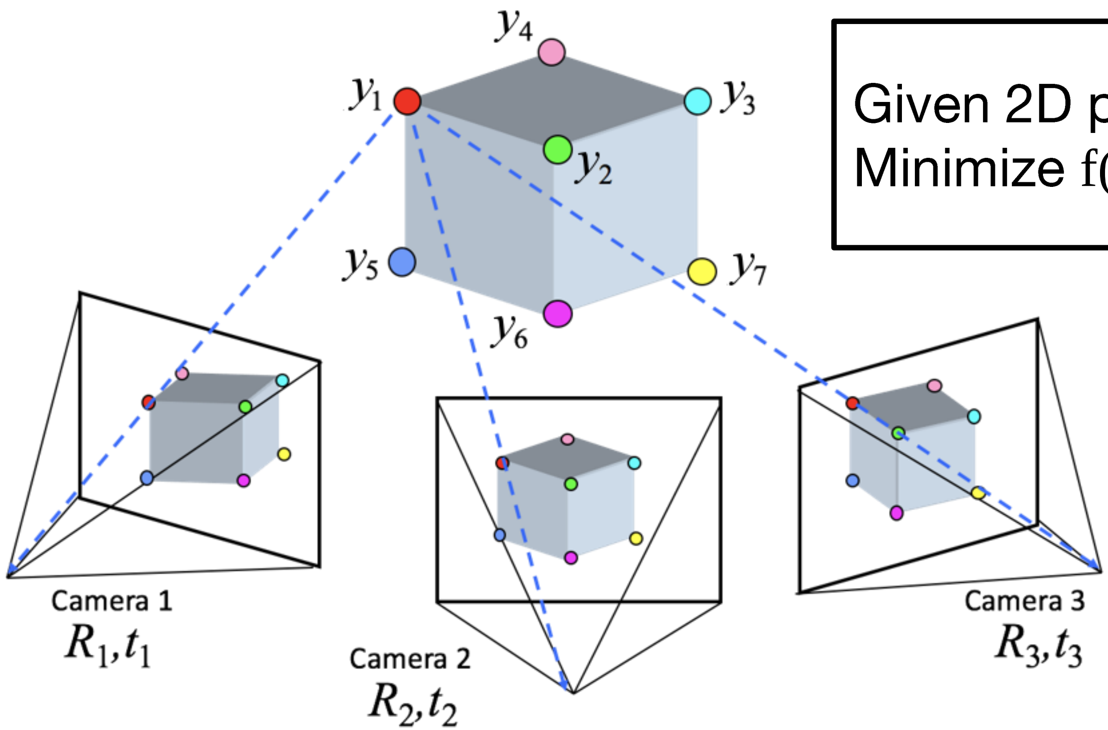
H. Strasdat 2012 PhD dissertation

Pose Estimation (Motion-only Bundle Adjustment)



Given 3D coordinates $\mathbf{Y}$
and 2D projections $\mathbf{Z}$
Minimize $f(\mathbf{R}, \mathbf{T}, \mathbf{Y})$

Structure from Motion



Given 2D projections $\mathbf{Z}$
Minimize $f(\mathbf{R}, \mathbf{T}, \mathbf{Y})$

Bundle Adjustment

$$\underset{\{(R,T)_i\},\{\mathbf{y}_j\}}{\operatorname{argmin}} \sum_{\mathbf{z}_{ij}} \| \overbrace{\pi(R_i^T(\mathbf{y}_j - T_i))}^{\hat{\mathbf{z}}_{ij}} - \mathbf{z}_{ij} \|_2^2$$

Reprojection of map point ***j*** in frame ***i***

Reprojection error for map point ***j*** in frame ***i***

Solve for poses $\{RT_i\}$ and map points $\{\mathbf{y}_j\}$ simultaneously

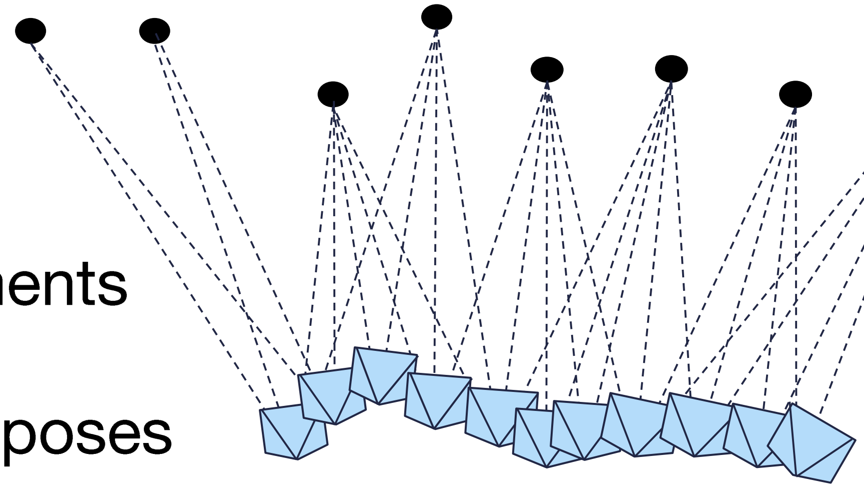
Measurement of map point ***j*** in frame ***i***

Full Optimization

Map points

Visual Measurements

Camera poses



$\mathbf{y}_j$

$\mathbf{z}_{ij}$

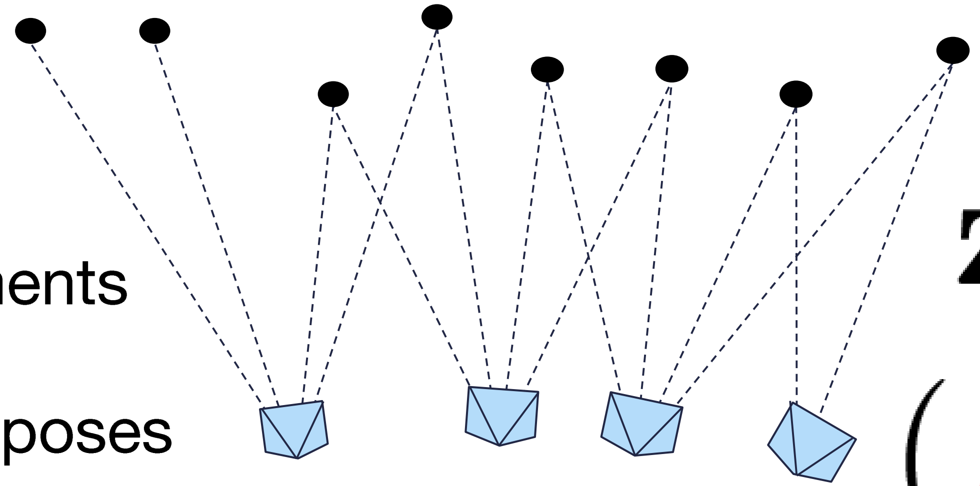
$(R, T)_i$

Keyframe Optimization

Map points

Visual Measurements

Camera poses



$\mathbf{y}_j$

$\mathbf{z}_{ij}$

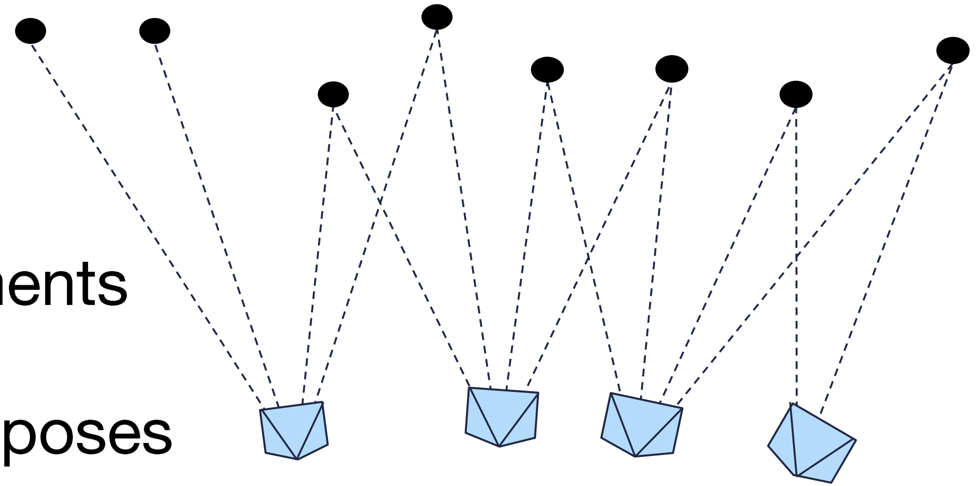
$(R, T)_i$

Scale Ambiguity

Map points

Visual Measurements

Camera poses



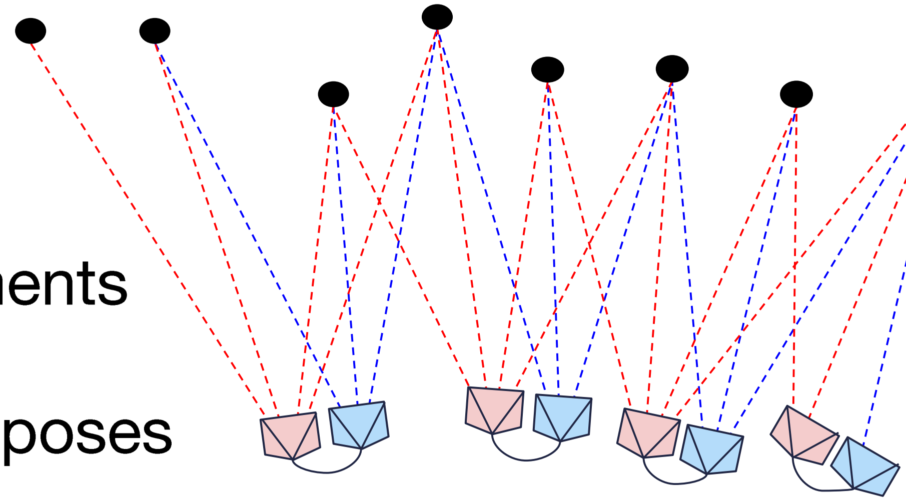
"You can't determine scale from purely angular constraints!"

Sparse Optimization with Binocular Priors

Map points

Visual Measurements

Camera poses



Fixed baseline

$$\Delta RT^{\mathrm{stereo}}$$

Bundle Adjustment with Binocular Prior

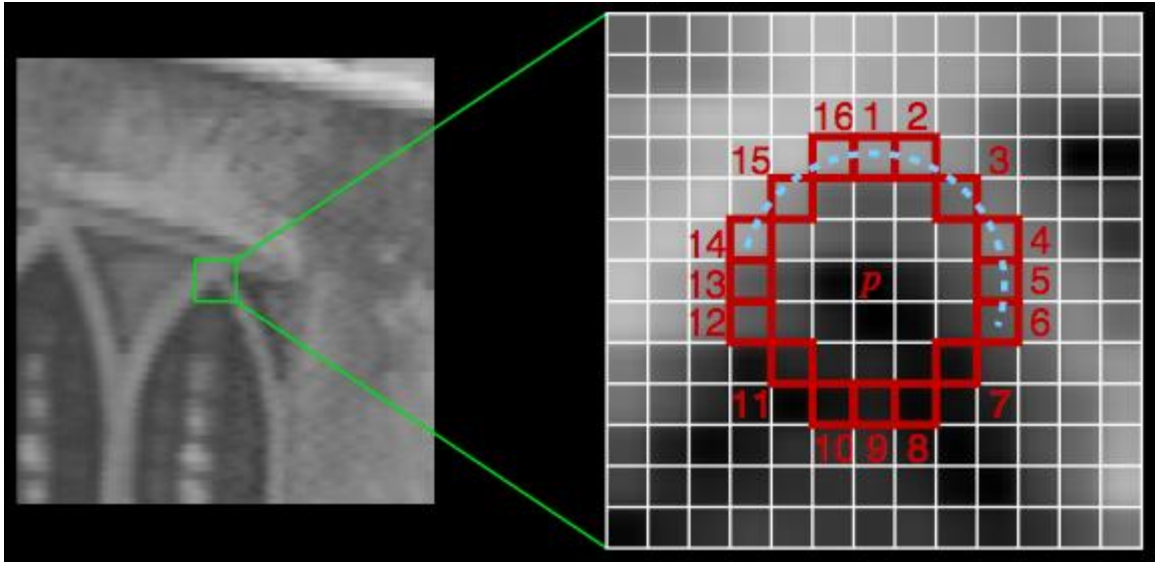
$$\underset{\{(R,T)_i\},\{\mathbf{y}_j\}}{\operatorname{argmin}} \sum_{\mathbf{z}_{ij}} \|\hat{\mathbf{z}}_{ij} - \mathbf{z}_{ij}\|_2^2$$

$$+ \lambda \sum_i d_{SE(3)}\left(\Delta RT_i^{\text{stereo}}, \Delta RT^{\text{stereo}}\right)$$

On-manifold error between estimated and calibrated baseline

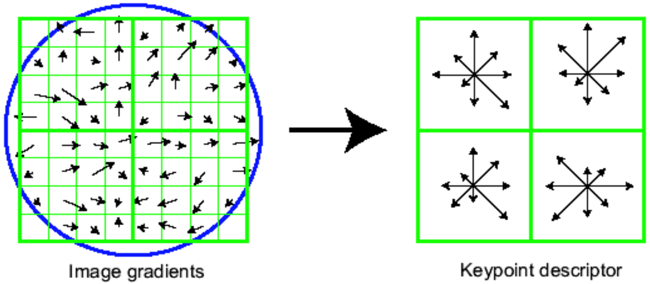
FAST Corners

- Look for a contiguous arc of N pixels
    - All much darker (or brighter) than the center pixel

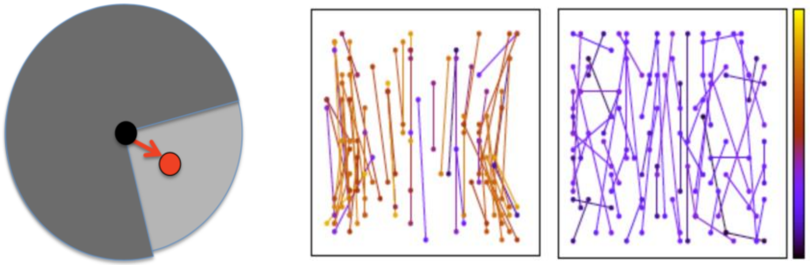


Matching Images

- With wide-baseline matching, template matching is not enough
  - The features look different - need **invariance** to viewpoint, illumination, …
  - Need efficient matching to a database previously seen features with ability to **discriminate** which is the right one



SIFT



ORB

Example: ORB-SLAM

Three threads running in parallel

- Tracking
- Local Mapping
- Loop Closing

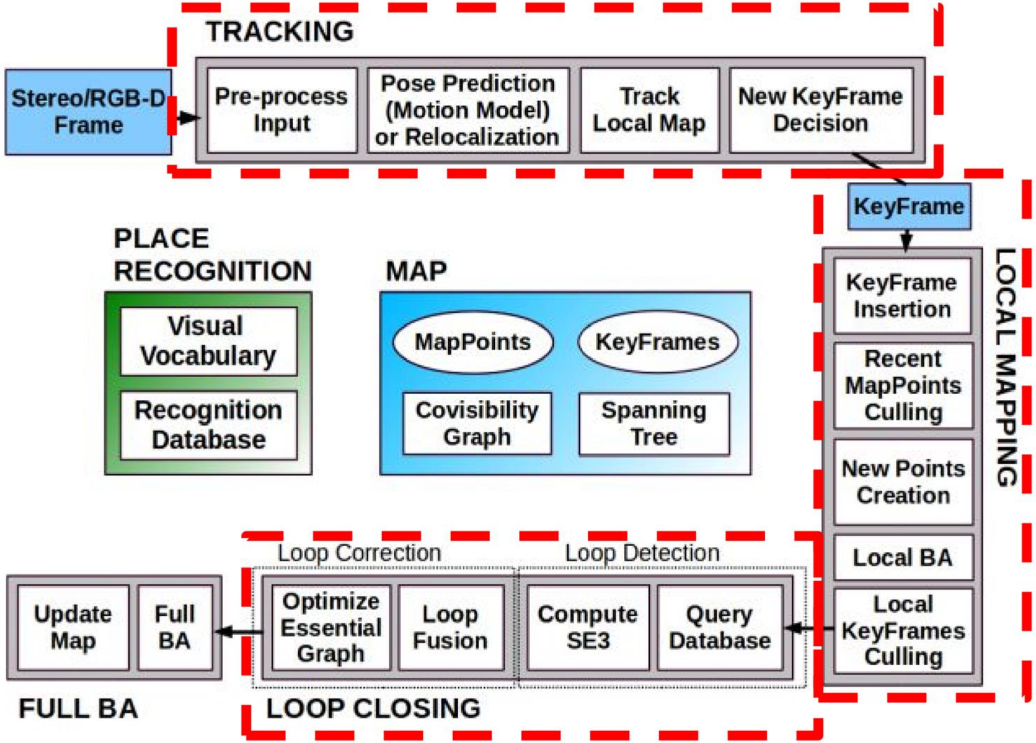
Feature-based
(Image is reduced to a sparse set of features)

Advantages:

- Wide baseline matching
- Illumination invariance
- Robust

Disadvantages:

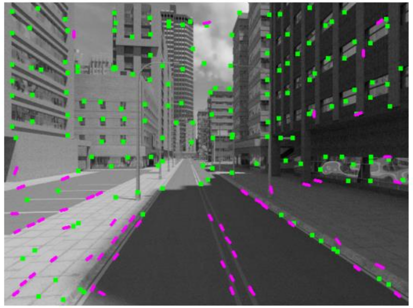
- Bad for efficiency and battery life

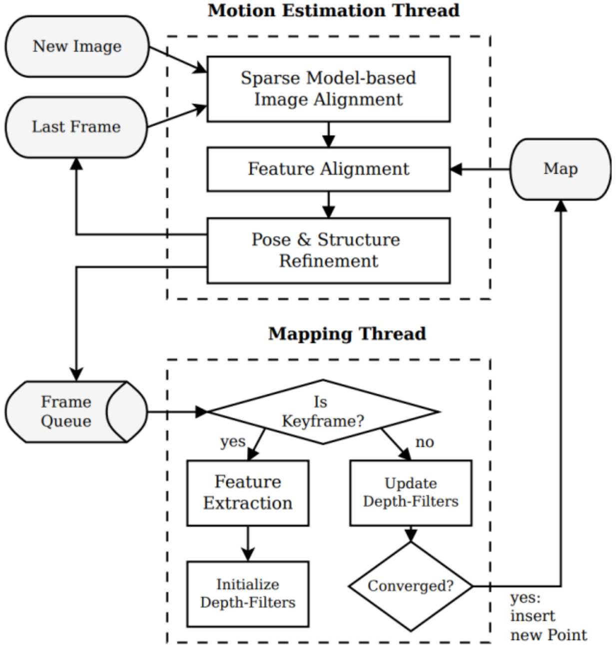


Example: Semi-Direct Visual Odometry (SVO)

- Visual Odometry - no loop closing or relocalization
- Two threads: motion estimation and mapping
- Edge and corner features
- Uses image intensities directly

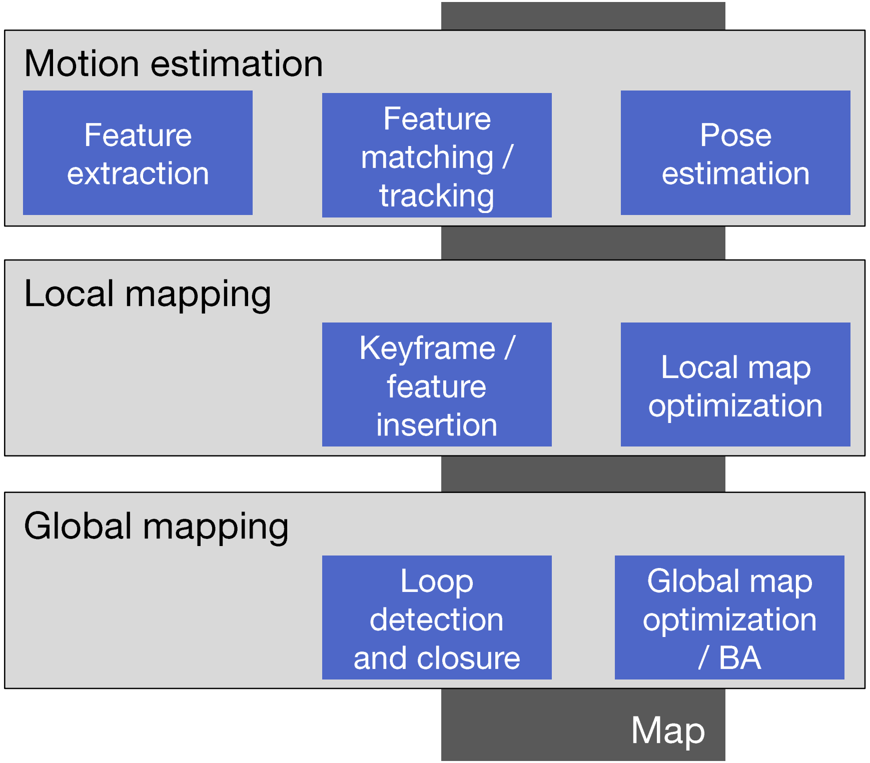
- Paper reports <50% CPU usage compared to ORB-SLAM





http://rpg.ifi.uzh.ch/svo2.html

Recap: Poses from Visual SLAM



Poses from Visual SLAM - Notes

- Pros:
  - Significantly lower drift than IMU (especially with stereo SLAM)
  - Relocalization
  - Map can be used for other computer vision and HCI purposes
- Cons:
  - High latency
  - Low frequency
  - Sensitive to environment
    - Lighting
    - Repetitive textures
    - Motion in the scene
  - Scale ambiguity in case of monocular SLAM

=> Inertial sensors provide complementary information

Outline

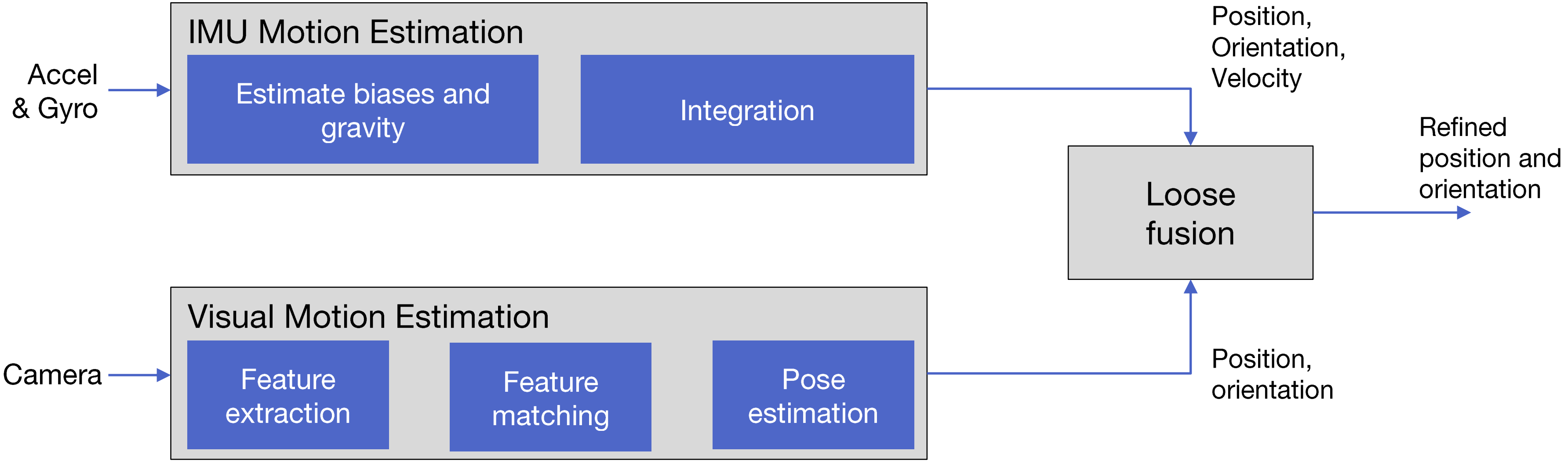
1. Pose Tracking for Augmented Reality Applications
2. Inertial Sensors and Pose Estimation
3. Introduction to Visual SLAM
4. Visual-Inertial Fusion
5. Applications and Future Challenges

Sensor Fusion

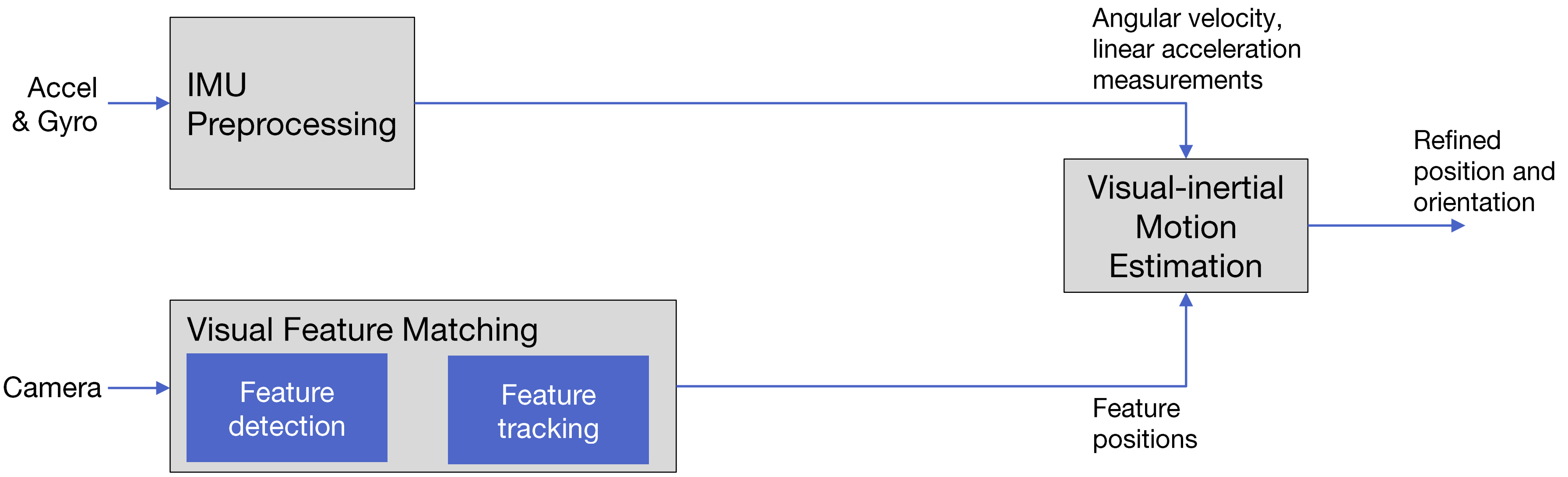
- Visual tracking and inertial sensors provide complementary information
- Sensor fusion should lead to better quality than either modality alone

- Loosely coupled fusion
  - Treat visual and inertial as black boxes and fuse at output level
- Tightly coupled fusion
  - Fuse information at sensor measurement level

Loosely Coupled Visual-Inertial Fusion



Tightly Coupled Visual-Inertial Fusion

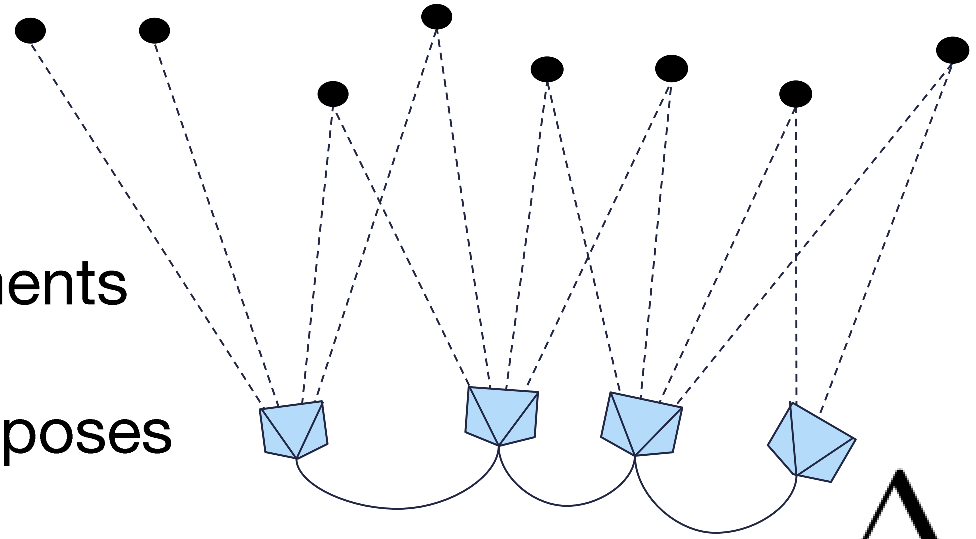


Bundle Adjustment Based Tight Fusion

Map points

Visual Measurements

Camera poses



Inertial priors

$$\Delta RT_i^{\mathrm{IMU}}$$

Bundle Adjustment with Inertial Prior

$$\underset{\{(R,T)_i\},\{\mathbf{y}_j\}}{\operatorname{argmin}} \sum_{\mathbf{z}_{ij}} \|\hat{\mathbf{z}}_{ij} - \mathbf{z}_{ij}\|_2^2$$

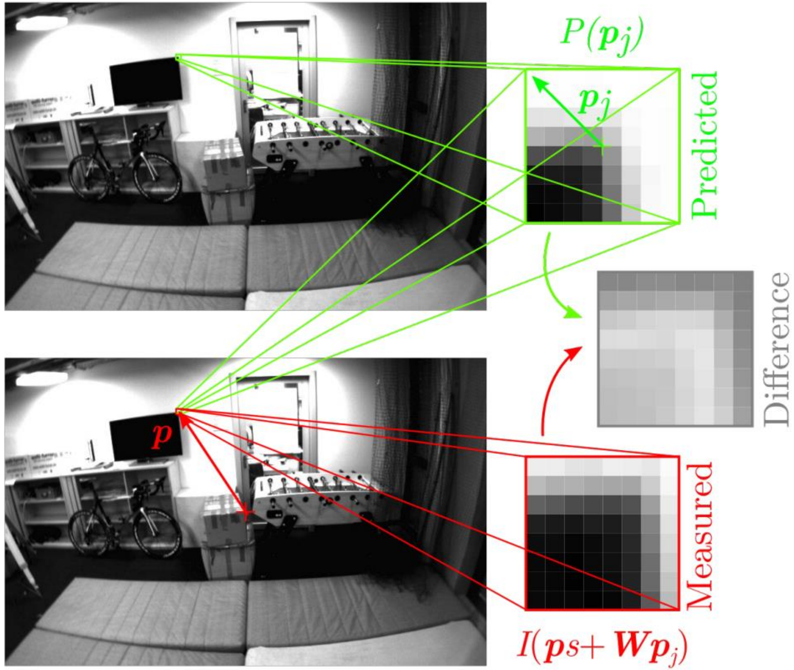
$$+ \lambda \sum_i d_{SE(3)} \left(\Delta RT_i, \Delta RT_i^{\mathrm{IMU}} \right)$$

On-manifold error between estimated and measured transitions

Tight Fusion / Filtering Example: ROVIO

- Extended Kalman Filter based IMU-Vision fusion
- State:
    - Position
    - Rotation
    - Velocity
    - IMU Biases
    - IMU-Camera extrinsics
    - Bearing and distance to *K* landmarks
- Direct method
    - Uses intensity differences

https://www.youtube.com/watch?v=ZMAISVy-6ao



https://github.com/ethz-asl/rovio

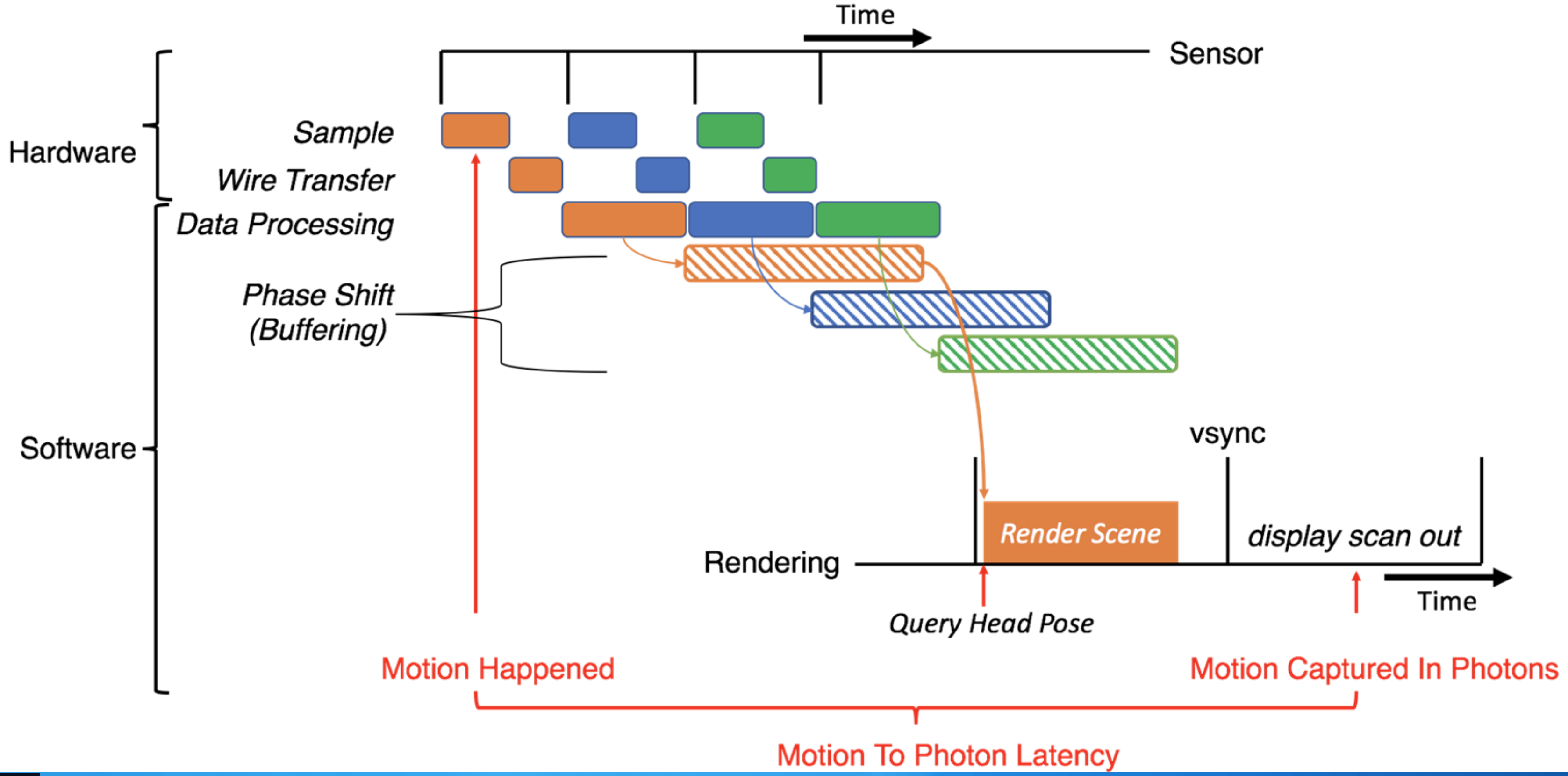
Outline

1. Pose Tracking for Augmented Reality Applications
2. Inertial Sensors and Pose Estimation
3. Introduction to Visual SLAM
4. Visual-Inertial Fusion
5. Applications and Future Challenges

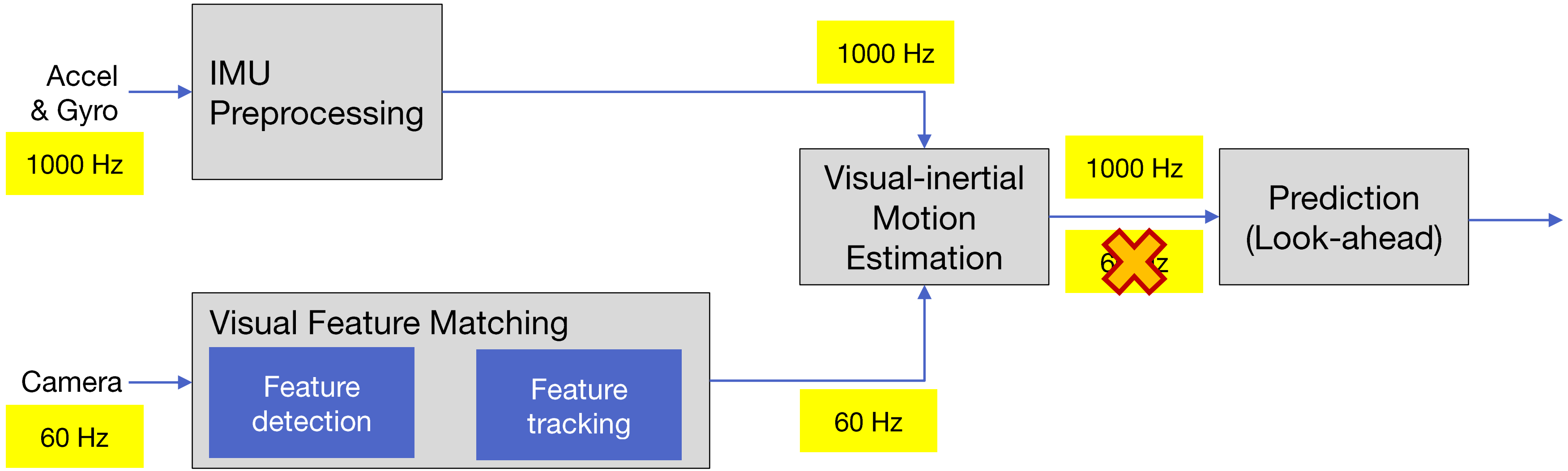
Low-latency Tracking and Rendering for AR

- What is latency?
  - Objects swim in optical-see-thru
  - Everything lags in video-see-thru -> nausea
- What causes it?
  - The capture – process – render – display cycle
- Solutions?
  - Be faster – in tracking and rendering
  - Predict (but prediction is difficult, and has its own problems)

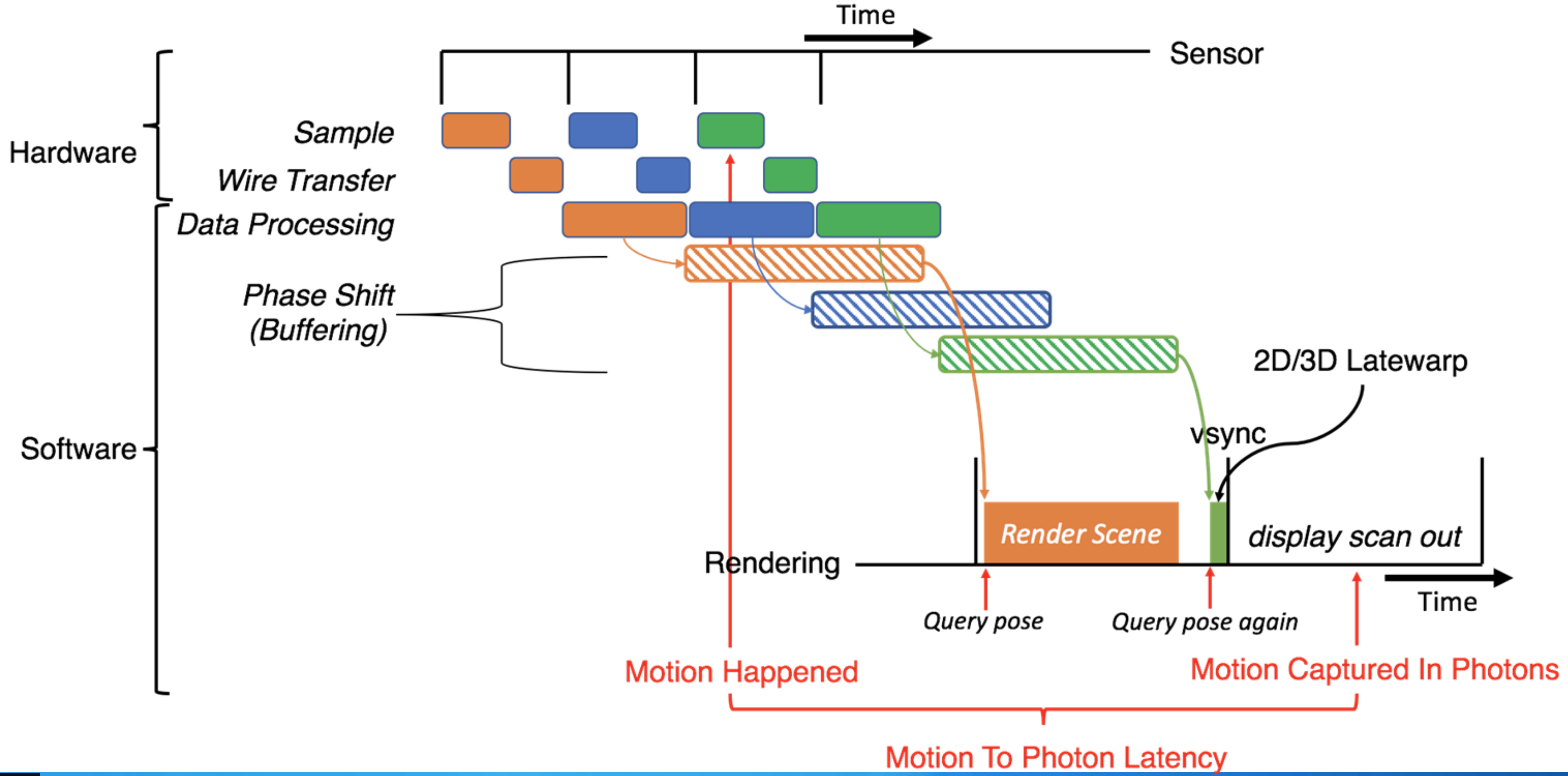
Motion-to-Photon Latency



Low-latency Visual-Inertial Fusion



Asynchronous Rendering



Visual-Inertial Tracking for AR – Challenges

- Computational complexity
- Low jitter – low latency
- Scale & drift
- Difficult environments:
    - Dark
    - High dynamic range
    - Repetitive texture
    - Motion in the scene

- Collaborative mapping
- Semantic SLAM

Resources

Excellent SLAM survey paper – focus on visual SLAM:

C. Cadena *et. al.* **Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age**. IEEE Transactions on Robotics, 2016. https://slam-future.github.io/ https://doi.org/10.1109/TRO.2016.2624754

Recent benchmark of monocular visual-inertial odometry algorithms:

J. Delmerico, D. Scaramuzza: **A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots**. IEEE International Conference on Robotics and Automation (ICRA), 2018. http://rpg.ifi.uzh.ch/publications.html

Open-source visual and visual-inertial SLAM algorithms

- **ORB-SLAM2** Visual SLAM supporting monocular, stereo and RGBD cameras. https://github.com/raulmur/ORB_SLAM2
- **ROVIO & maplab** Visual-inertial, filtering-based tracking and mapping. https://github.com/ethz-asl/maplab
- **VINS-Mono** Monocular, optimization based visual-inertial SLAM including loop closing; also with mobile implementation. https://github.com/HKUST-Aerial-Robotics/VINS-Mono