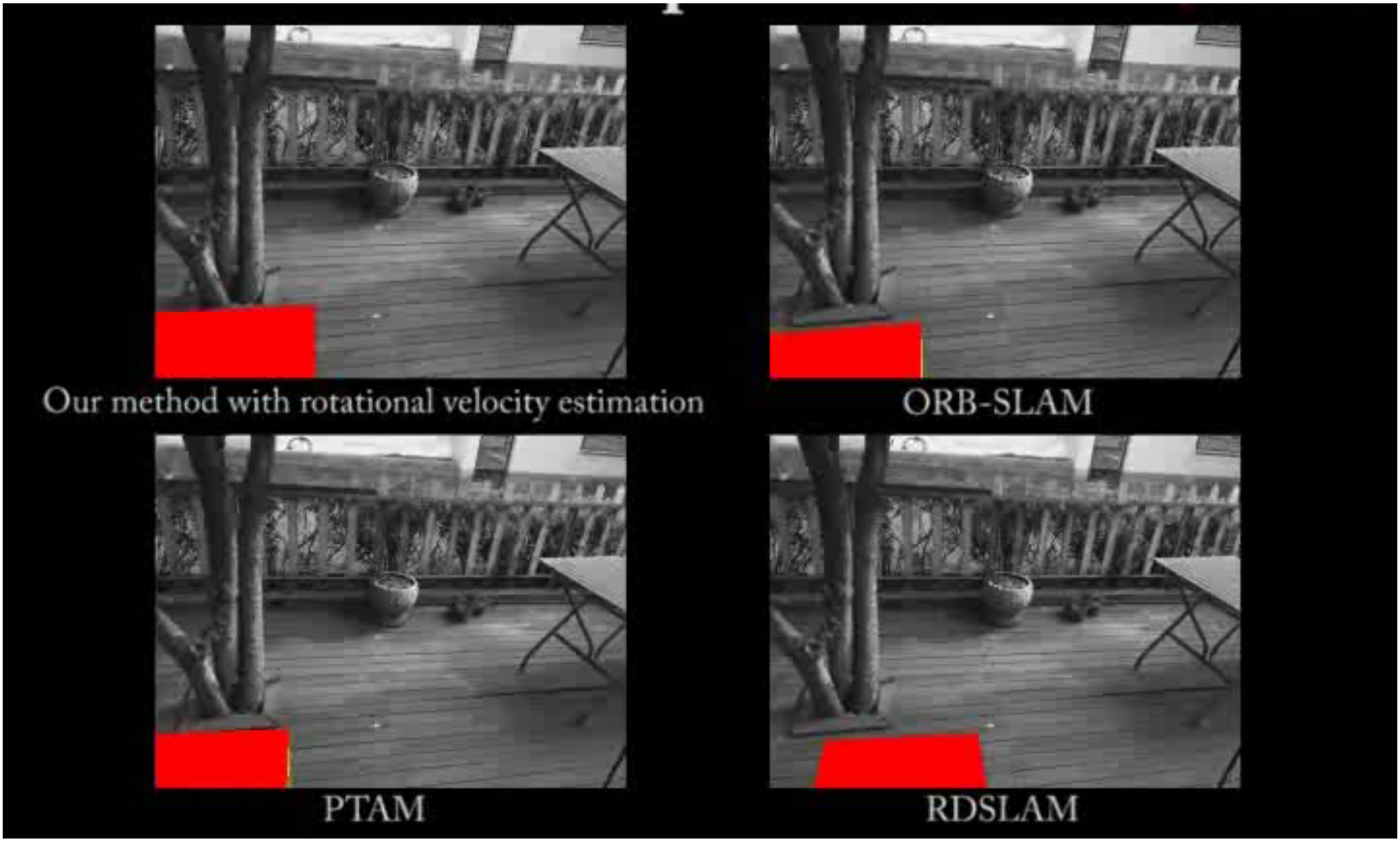
Visual SLAM for Mobile

Instructor - Simon Lucey

16-623 - Designing Computer Vision Apps

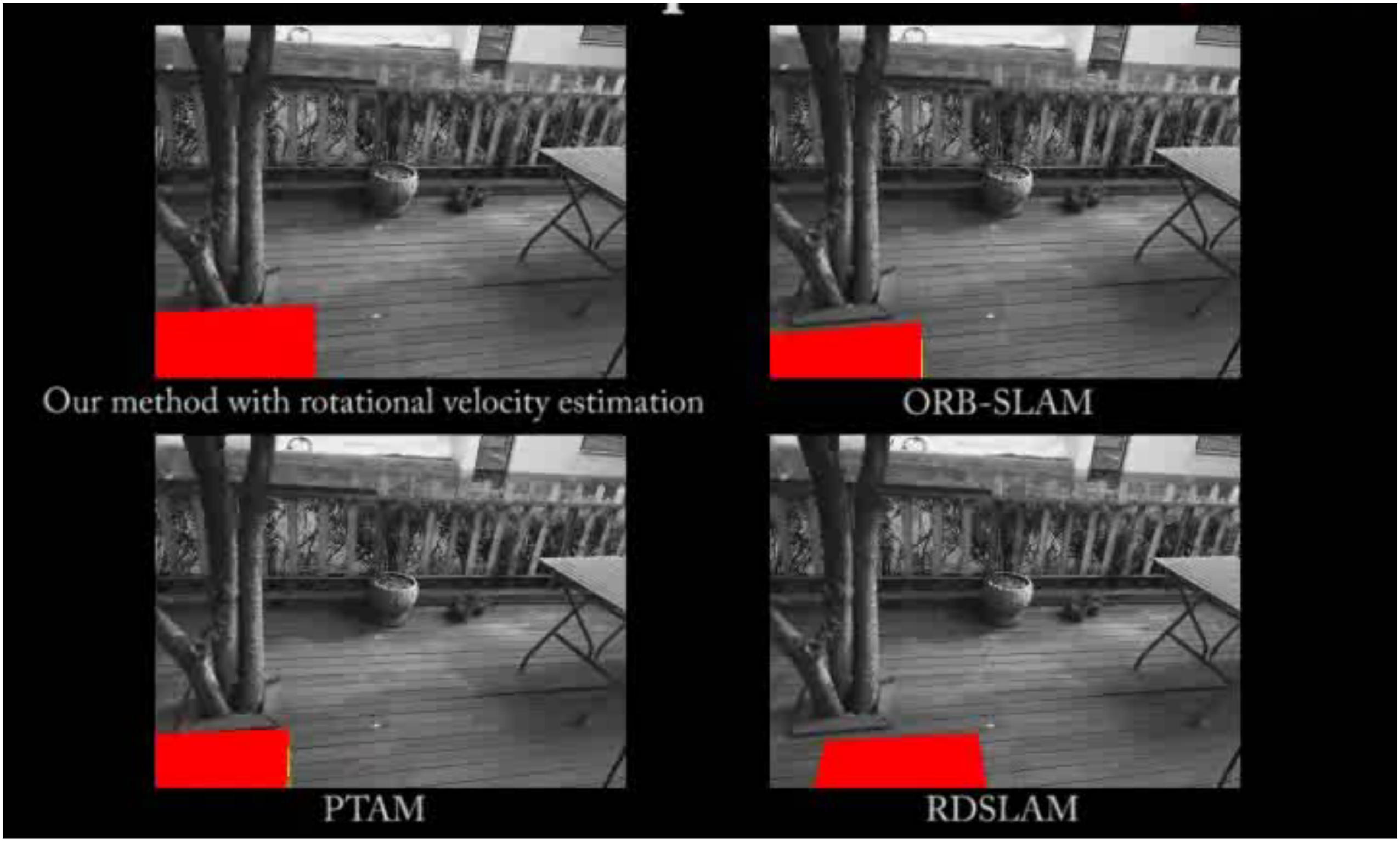
Carnegie Mellon
THE ROBOTICS INSTITUTE

Example of SLAM for AR



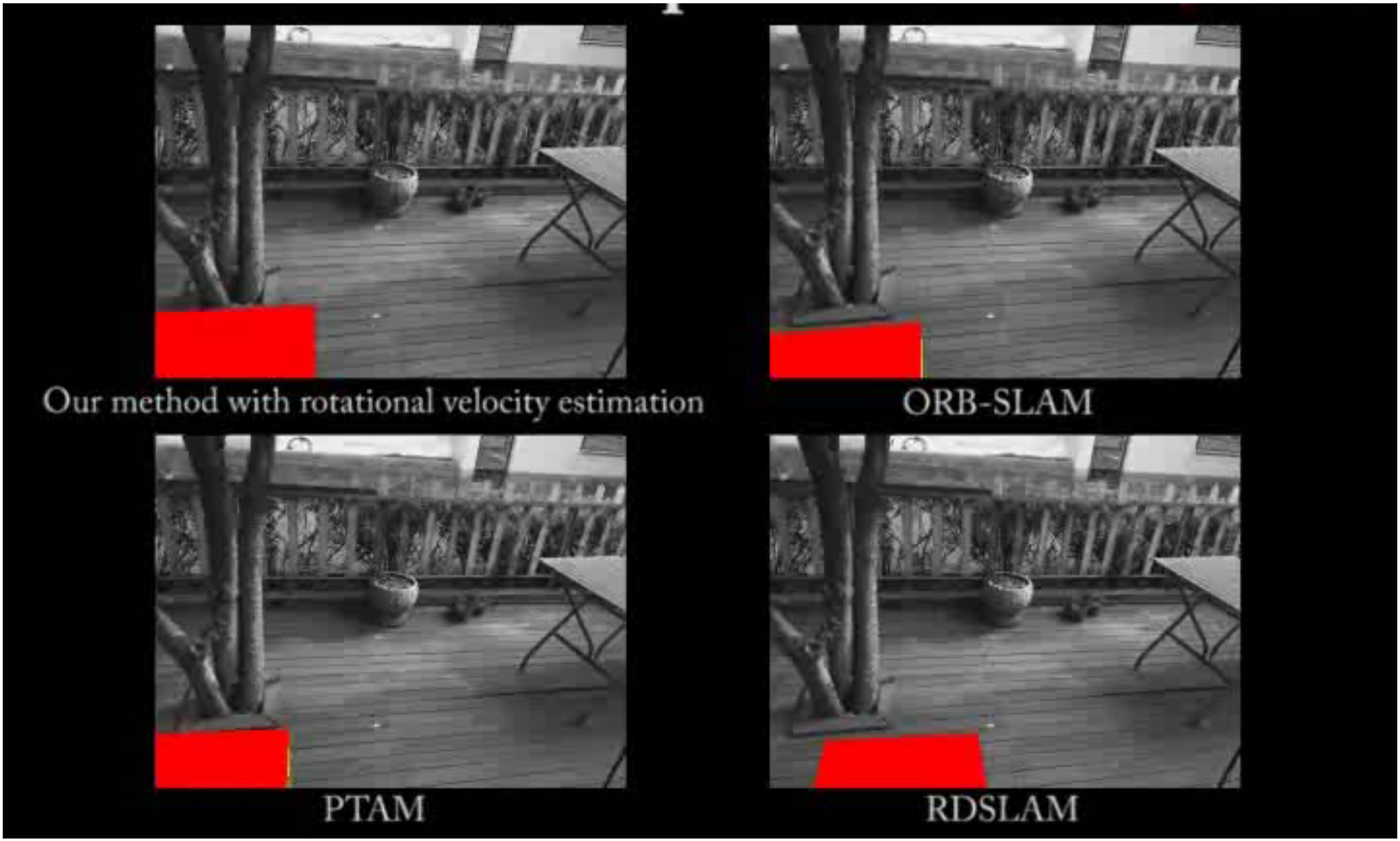
Taken from: H. Liu et al. “Robust Keyframe-based Monocular SLAM for Augmented Reality”, ISMAR 2016.

Example of SLAM for AR



Taken from: H. Liu et al. “Robust Keyframe-based Monocular SLAM for Augmented Reality”, ISMAR 2016.

Example of SLAM for AR



Taken from: H. Liu et al. “Robust Keyframe-based Monocular SLAM for Augmented Reality”, ISMAR 2016.

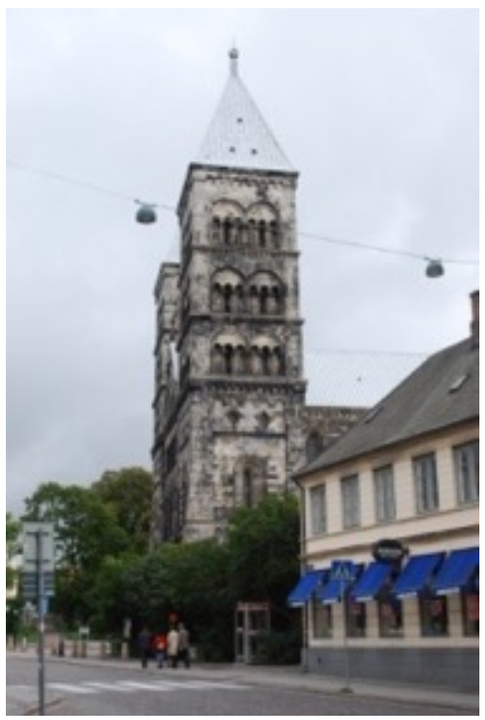
What is SLAM??

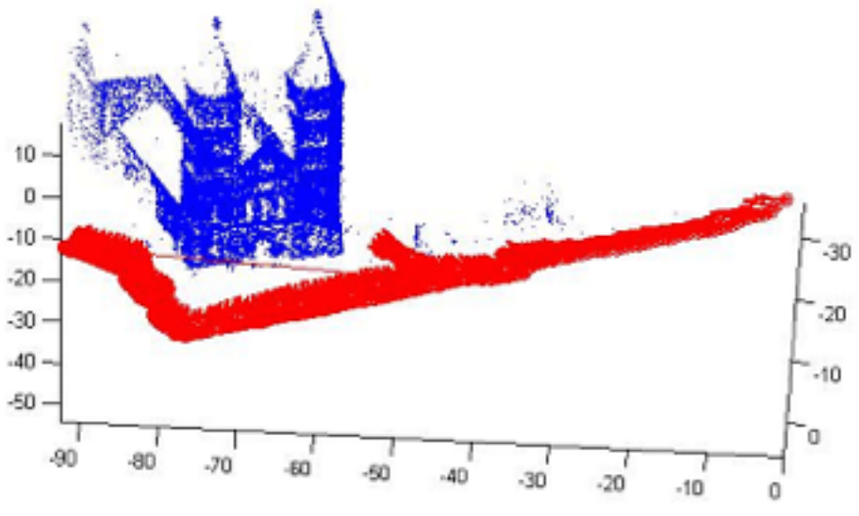
- **S**imultaneous **L**ocalization **a**nd **M**apping.
- On mobile interested primarily in Visual SLAM (VSLAM).
- Sometimes called Mono SLAM if there is only one camera.
- Can be viewed as an online SfM problem.

Today

- SfM - Bundle Adjustment
- VSLAM - Keyframe vs. Filtering
- Visual Odometry
- Loop Closure

Reminder - Bundle Adjustment





The cathedral dataset:

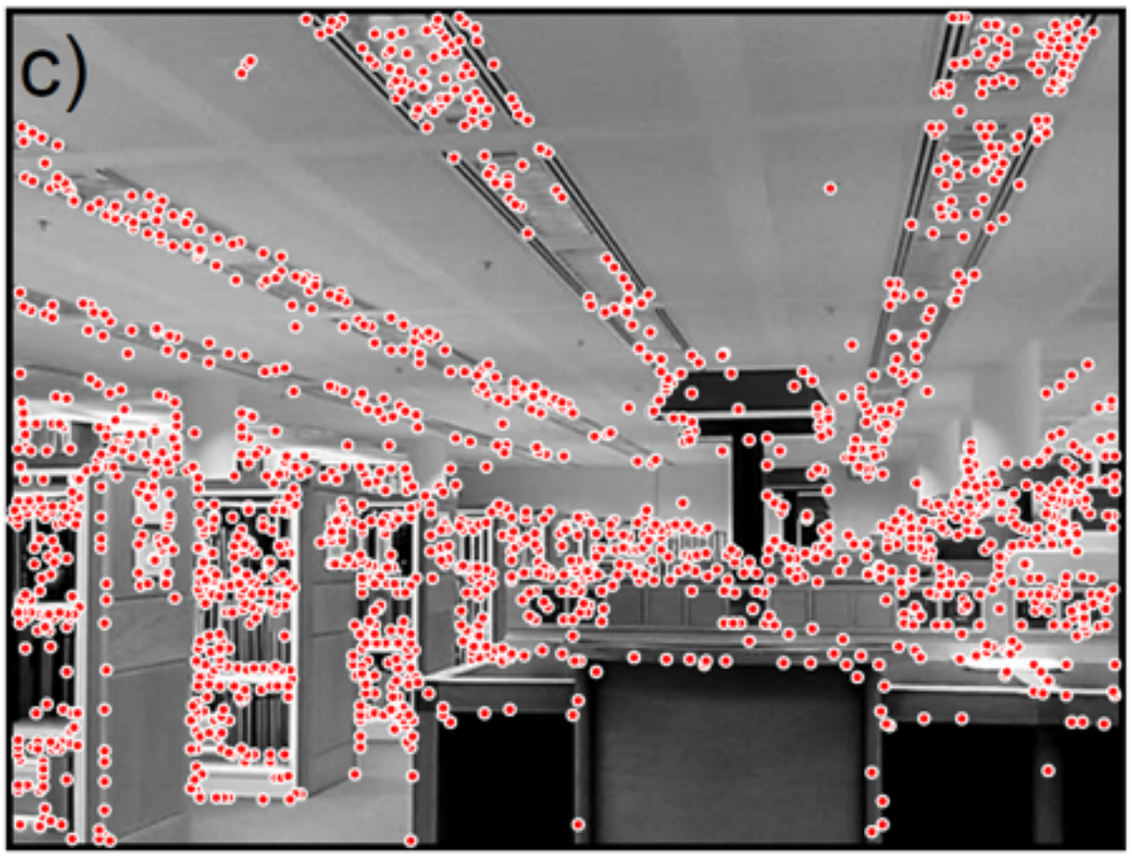
- 480 camera matrices $[\boldsymbol{\Omega}_i, \boldsymbol{\tau}_i]$
- Total dof = $480 \times (3+3) = 2880$
- 91178 3D points.
- Total dof = $91178 \times 3 = 273543$

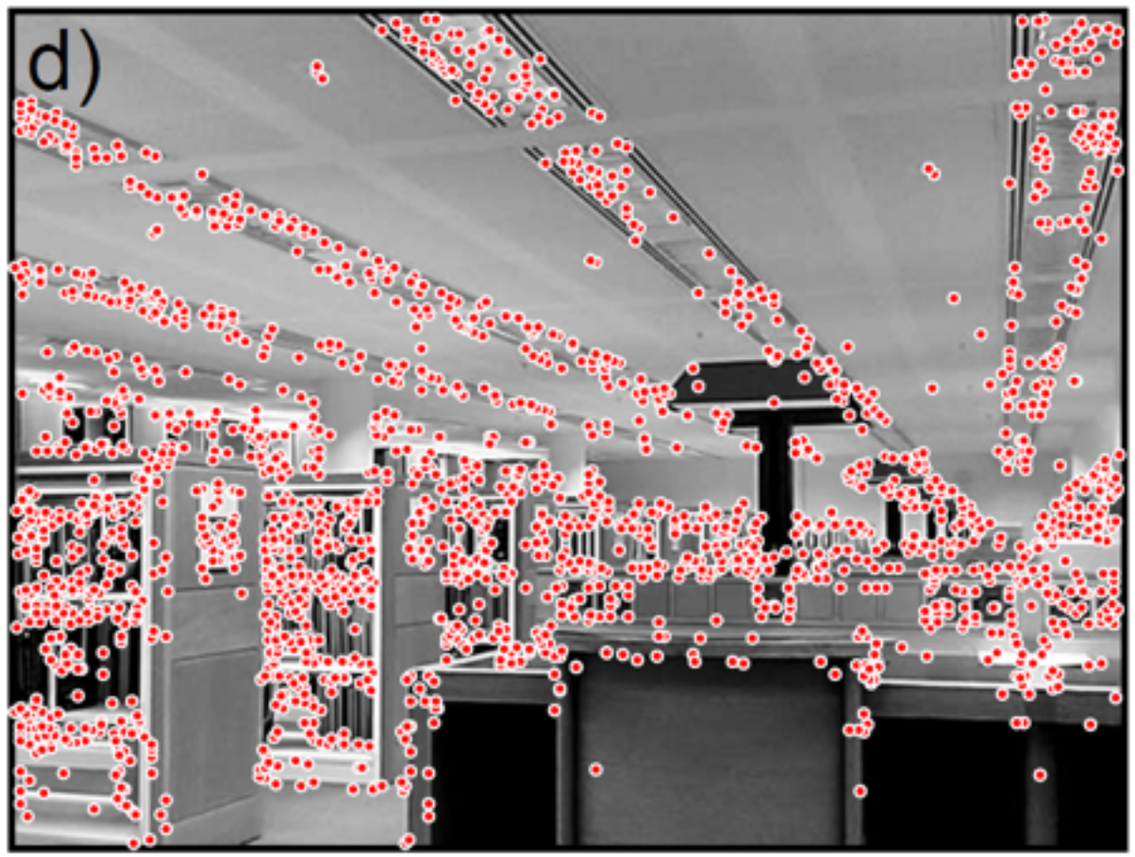
Adapted from: Optimization Methods in Computer Vision. Anders Eriksson

Reminder - Two view reconstruction

Start with pair of images taken from slightly different viewpoints

Reminder - Two view reconstruction



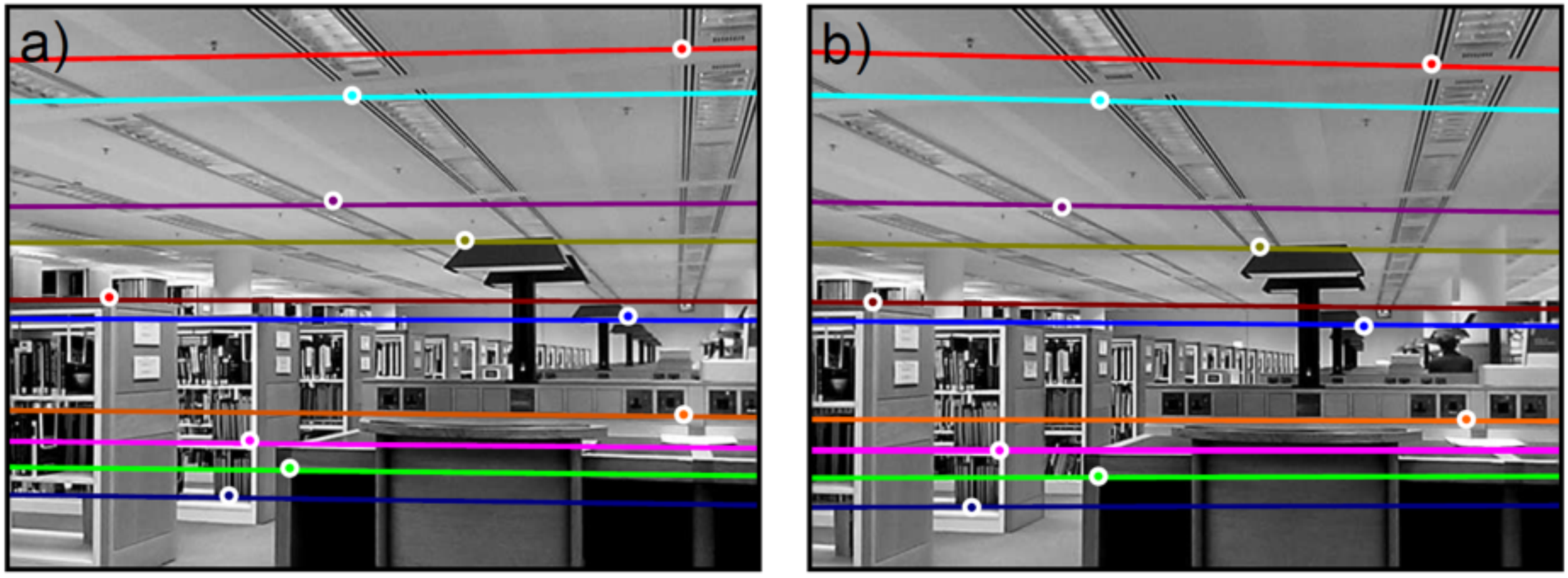


Find features using a corner detection algorithm

Reminder - Two view reconstruction

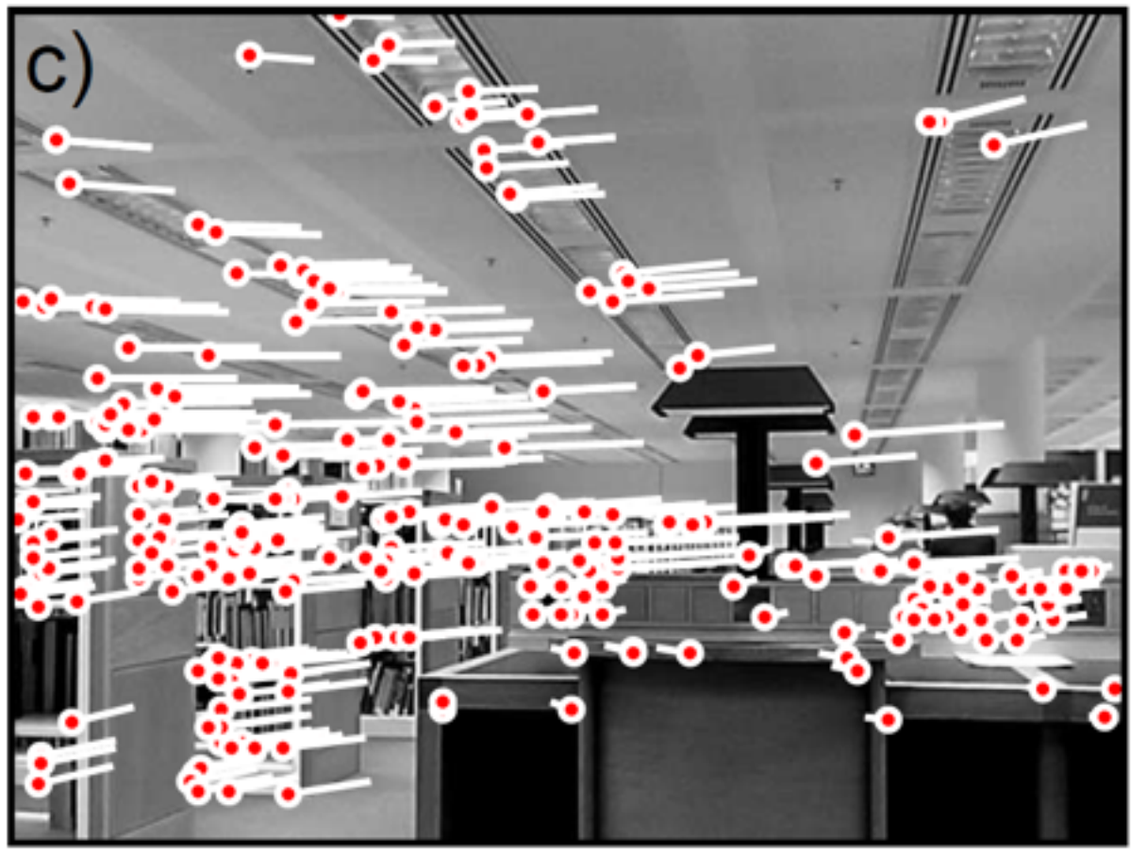
Match features using a greedy algorithm

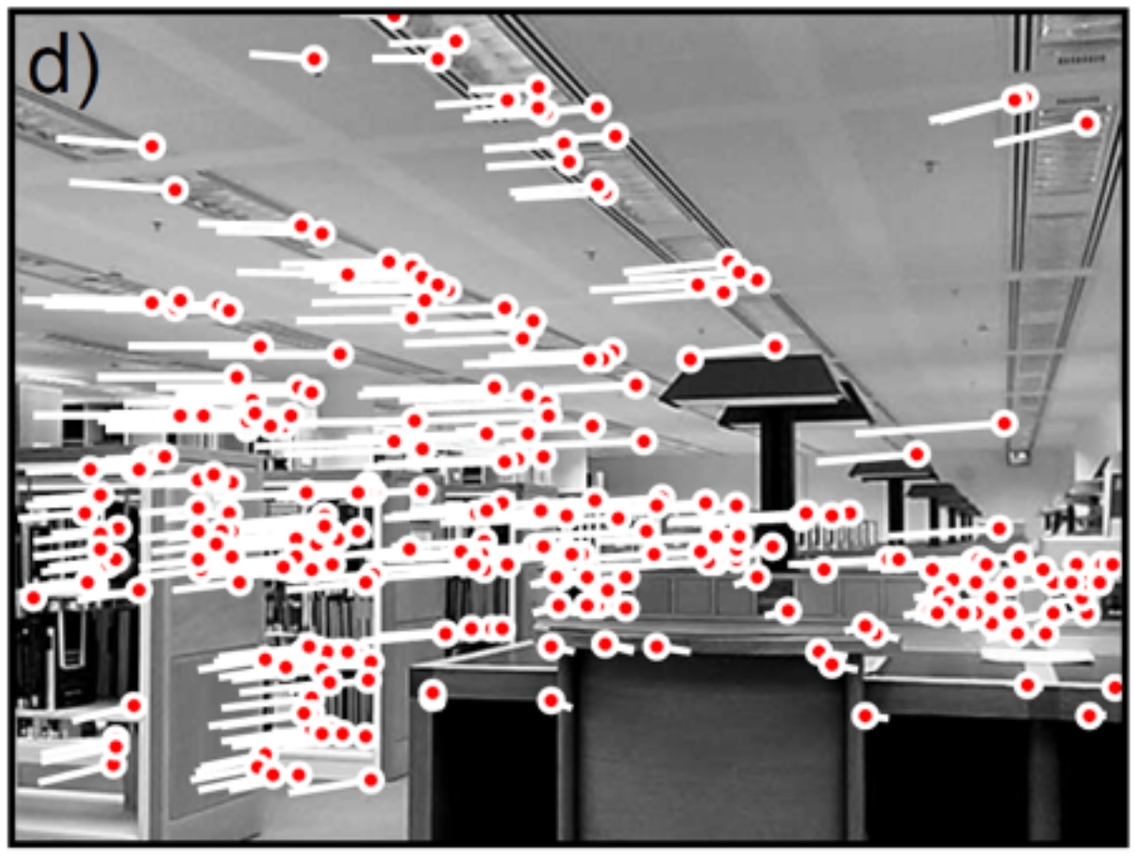
Reminder - Two view reconstruction



Fit fundamental matrix using robust algorithm such as RANSAC

Reminder - Two view reconstruction





Find matching points that agree with the fundamental matrix

Reminder - Two view reconstruction

- Extract essential matrix from fundamental matrix.
- Extract $\boldsymbol{\Omega}$ rotation and $\boldsymbol{\tau}$ translation from essential matrix.
- Reconstruct the 3D positions $\mathbf{w}$ of points.

$$\lambda\tilde{\mathbf{x}} = \boldsymbol{\Omega}\mathbf{w} + \boldsymbol{\tau}$$

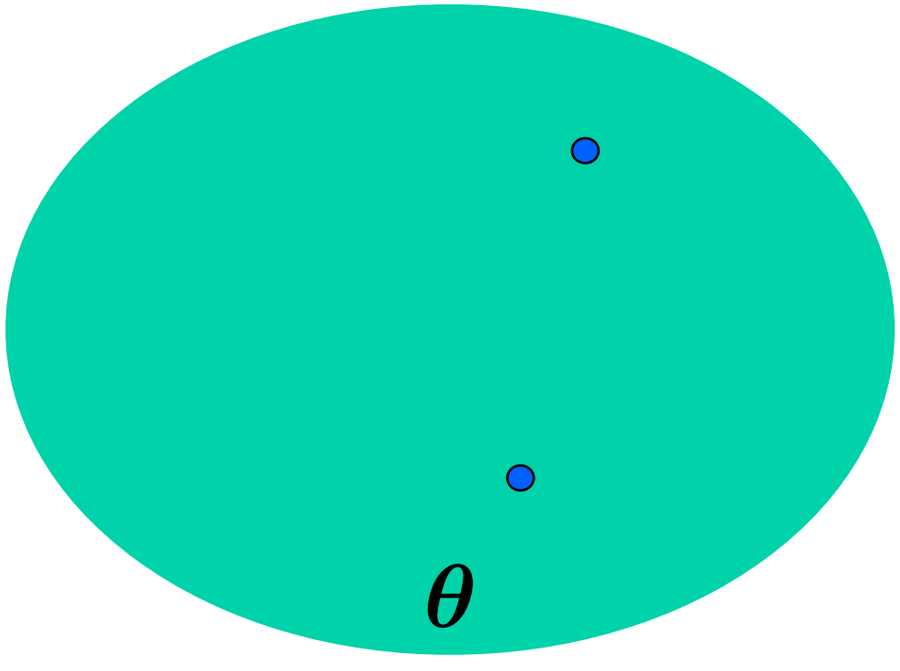
- We refer to these matrices as belonging to the Special Euclidean Group - SE(3).

$$\mathbf{T} = \begin{bmatrix} \boldsymbol{\Omega} & \boldsymbol{\tau} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

Reminder: Lie Algebra

- Exponential maps on the SO(3), SL(3) and SE(3) groups are related to the much broader topic of Lie Algebra.
- More details on this topic can be found at in Murray et al. 1994.

$$\mathbf{T} = \begin{bmatrix} \boldsymbol{\Omega} & \boldsymbol{\tau} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

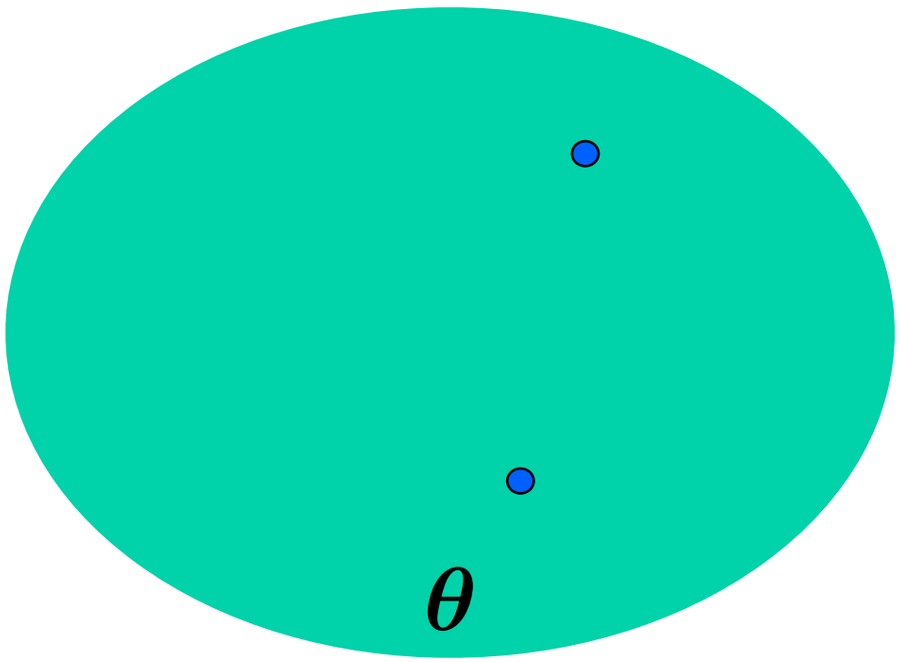


"Sophus Lie"

Reminder: Lie Algebra

- Exponential maps on the SO(3), SL(3) and SE(3) groups are related to the much broader topic of Lie Algebra.
- More details on this topic can be found at in Murray et al. 1994.

$$\mathbf{T}(\boldsymbol{\theta}) = \exp\left(\sum_{i=1}^{6} \theta_i \mathbf{A}_i\right) \in \mathrm{SE}(3)$$

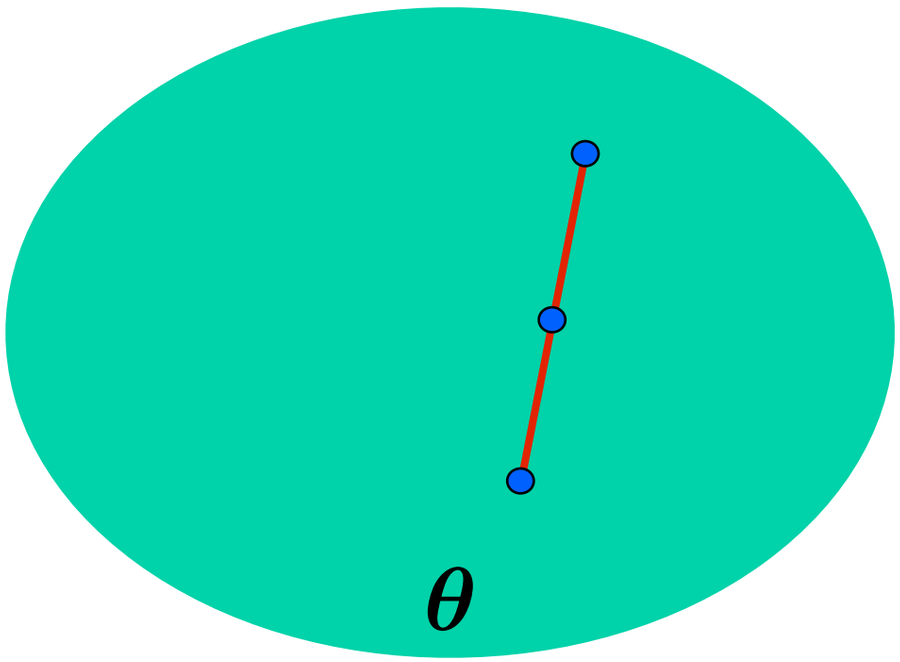


"Sophus Lie"

Reminder: Lie Algebra

- Exponential maps on the SO(3), SL(3) and SE(3) groups are related to the much broader topic of Lie Algebra.
- More details on this topic can be found at in Murray et al. 1994.

$$\mathbf{T}(\boldsymbol{\theta}) = \exp\left(\sum_{i=1}^{6} \theta_i \mathbf{A}_i\right) \in \mathrm{SE}(3)$$



"Sophus Lie"

SfM - Bundle Adjustment

$$\arg\min_{\mathbf{w},\boldsymbol{\theta}} \sum_{f=1}^{F} \sum_{n=1}^{N} ||\mathbf{x}_n^f - \pi(\mathbf{w}_n; \boldsymbol{\theta}^f)||_2^2$$

$\mathbf{x} \leftarrow$ 2D projection $\mathbf{w} \leftarrow$ 3D point

$\boldsymbol{\theta} \leftarrow$ extrinsics $N \leftarrow$ no. of points

$\pi \leftarrow$ projection function

$F \leftarrow$ no. of frames

SfM - Linearization

$$\pi(\mathbf{w}_n + \Delta\mathbf{w}_n; \boldsymbol{\theta}_f \circ \Delta\boldsymbol{\theta}_f) \approx \pi(\mathbf{w}_n; \boldsymbol{\theta}_f) + \mathbf{J}_n^f \begin{bmatrix} \Delta\boldsymbol{\theta}_f \\ \Delta\mathbf{w}_n \end{bmatrix}$$

SfM - Linearization

$$\pi(\mathbf{w}_n + \Delta\mathbf{w}_n; \boldsymbol{\theta}_f \odot \Delta\boldsymbol{\theta}_f) \approx \pi(\mathbf{w}_n; \boldsymbol{\theta}_f) + \mathbf{J}_n^f \begin{bmatrix} \Delta\boldsymbol{\theta}_f \\ \Delta\mathbf{w}_n \end{bmatrix}$$

why not additive??

SfM - Linearization

$$\pi(\mathbf{w}_n + \Delta\mathbf{w}_n; \boldsymbol{\theta}_f \circ \Delta\boldsymbol{\theta}_f) \approx \pi(\mathbf{w}_n; \boldsymbol{\theta}_f) + \mathbf{J}_n^f \begin{bmatrix} \Delta\boldsymbol{\theta}_f \\ \Delta\mathbf{w}_n \end{bmatrix}$$

$$\arg \min_{\Delta\boldsymbol{\theta}, \Delta\mathbf{w}} \sum_{f=1}^{F} \sum_{n=1}^{N} ||\mathbf{x}_n^f - \pi(\mathbf{w}_n; \boldsymbol{\theta}_f) - \mathbf{J}_n^f \begin{bmatrix} \Delta\boldsymbol{\theta}_f \\ \Delta\mathbf{w}_n \end{bmatrix} ||_2^2$$

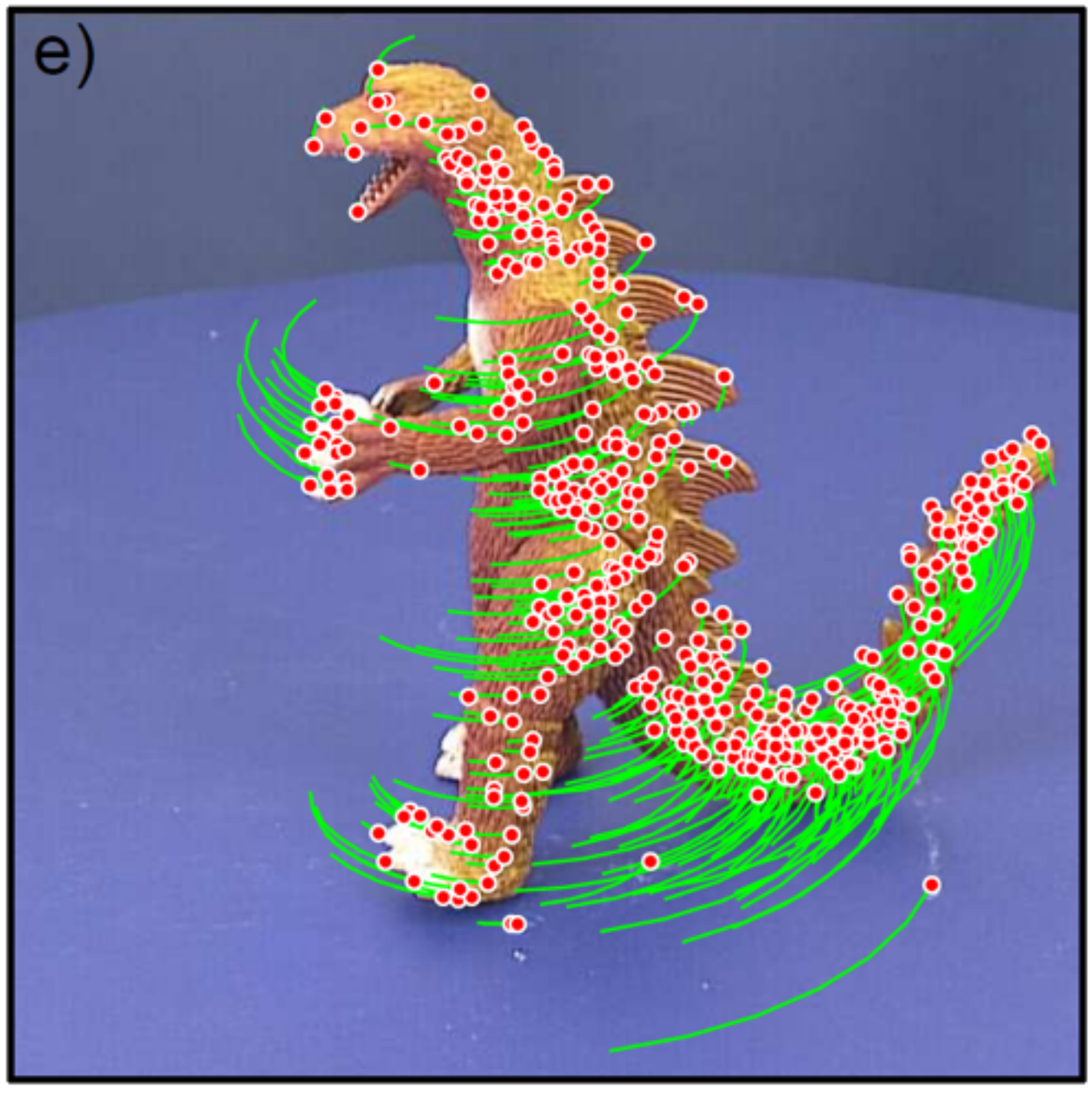
$\mathbf{x} \leftarrow$ 2D projection $\mathbf{w} \leftarrow$ 3D point

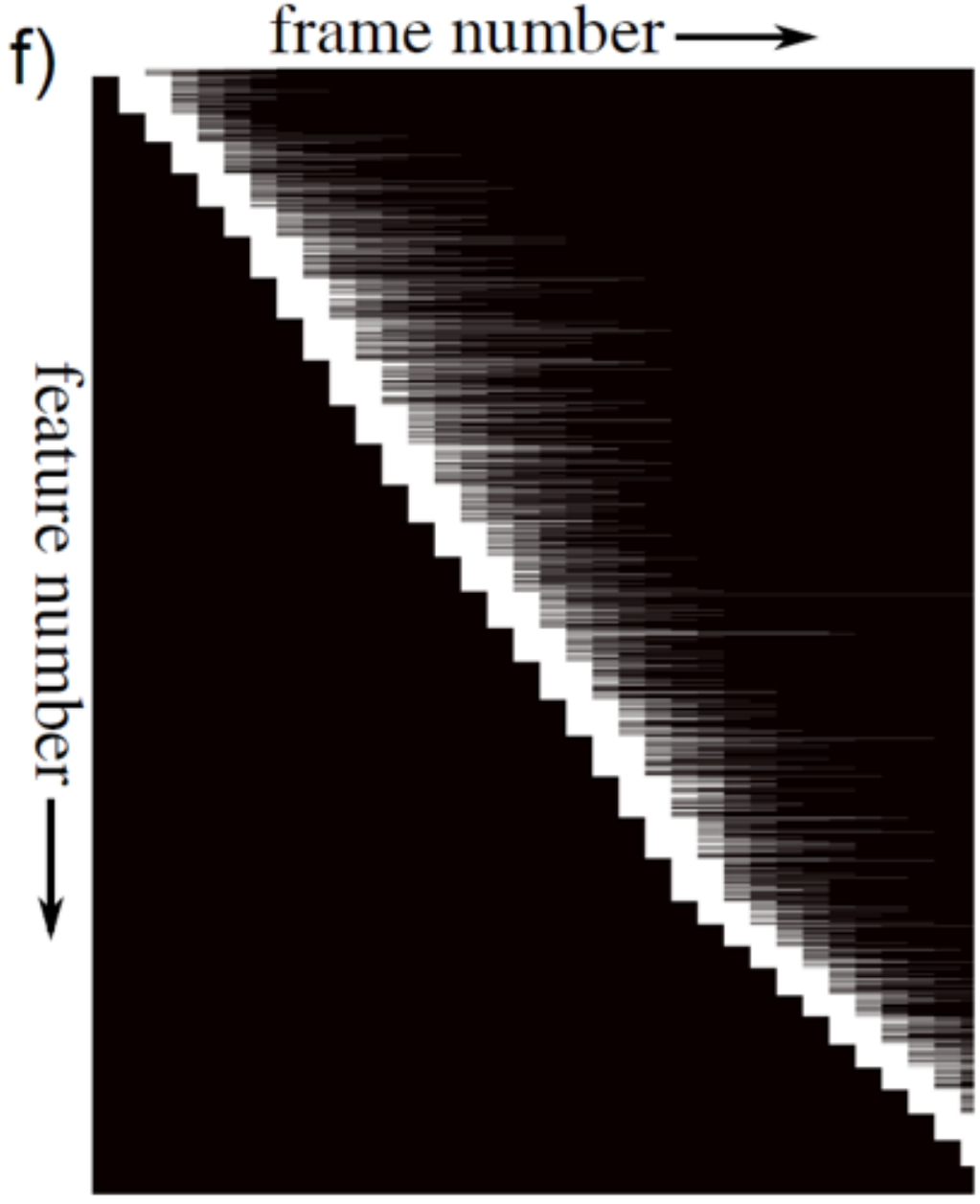
$\boldsymbol{\theta} \leftarrow$ extrinsics $N \leftarrow$ no. of points

$\pi \leftarrow$ projection function

$F \leftarrow$ no. of frames

Visibility of Points





$$\Upsilon = \begin{bmatrix} \rho_1^1 & \cdots & \rho_1^F \\ \vdots & \ddots & \vdots \\ \rho_N^1 & \cdots & \rho_N^F \end{bmatrix}$$

"visibility matrix"

SfM - Bundle Adjustment

$$\pi(\mathbf{w}_n + \Delta\mathbf{w}_n; \boldsymbol{\theta}_f \circ \Delta\boldsymbol{\theta}_f) \approx \pi(\mathbf{w}_n; \boldsymbol{\theta}_f) + \mathbf{J}_n^f \begin{bmatrix} \Delta\boldsymbol{\theta}_f \\ \Delta\mathbf{w}_n \end{bmatrix}$$

$$\arg \min_{\Delta\boldsymbol{\theta},\Delta\mathbf{w}} \sum_{f=1}^{F} \sum_{n=1}^{N} \rho_n^f ||\mathbf{x}_n^f - \pi(\mathbf{w}_n; \boldsymbol{\theta}_f) - \mathbf{J}_n^f \begin{bmatrix} \Delta\boldsymbol{\theta}_f \\ \Delta\mathbf{w}_n \end{bmatrix} ||_2^2$$

$\mathbf{x} \leftarrow$ 2D projection $\mathbf{w} \leftarrow$ 3D point

$\boldsymbol{\theta} \leftarrow$ extrinsics $N \leftarrow$ no. of points

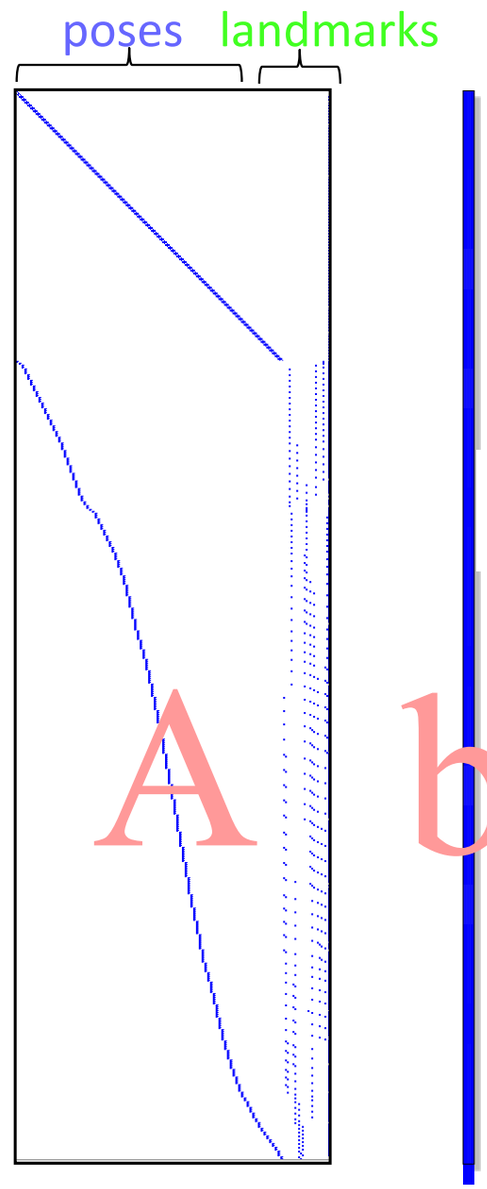
$\pi \leftarrow$ projection function

$F \leftarrow$ no. of frames $\rho \rightarrow$ visibility $\in [0, 1]$

SfM - Bundle Adjustment

$$\arg \min_{\Delta\boldsymbol{\theta},\Delta\mathbf{w}} ||\mathbf{b} - \mathbf{A} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{w} \end{bmatrix} ||_2^2$$

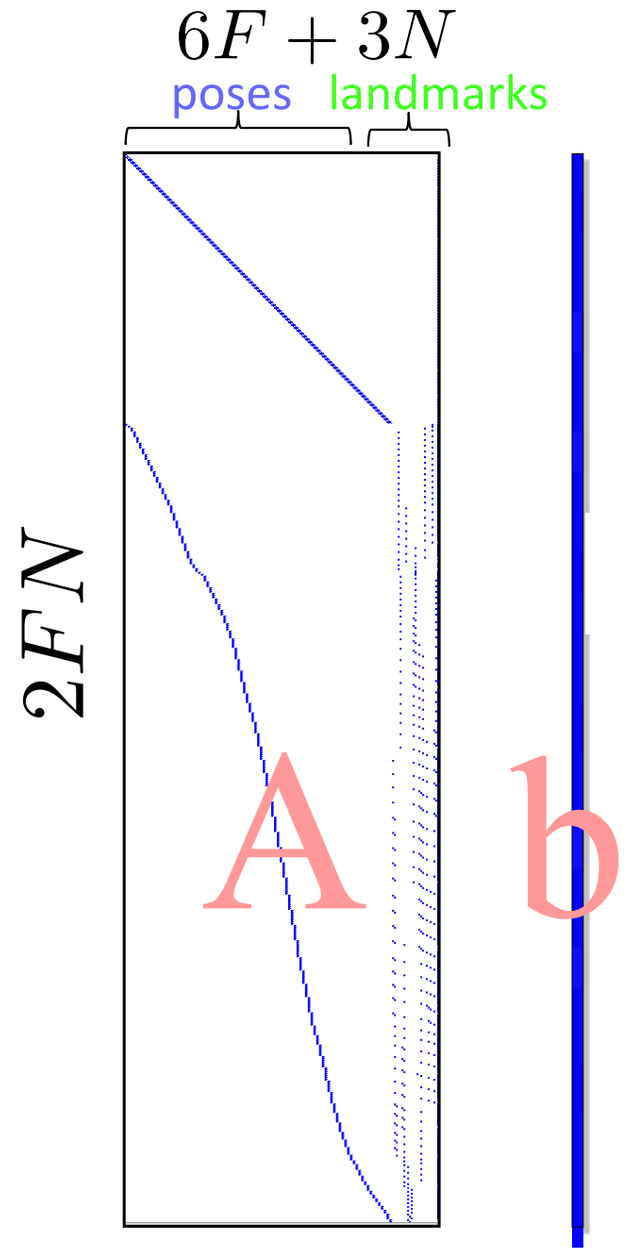
- Can be solved efficiently using sparse linear solvers such as,
  - Google Ceres Solver - http://ceres-solver.org
  - G2o - https://openslam.org/g2o.html .
- Then iteratively apply GN or LM algorithm.



SfM - Bundle Adjustment

$$\arg \min_{\Delta\boldsymbol{\theta},\Delta\mathbf{w}} ||\mathbf{b} - \mathbf{A} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{w} \end{bmatrix} ||_2^2$$

- Can be solved efficiently using sparse linear solvers such as,
  - Google Ceres Solver - http://ceres-solver.org
  - G2o - https://openslam.org/g2o.html .
- Then iteratively apply GN or LM algorithm.



Reminder: Gauss-Newton Algorithm

- Gauss-Newton (GN) algorithm common strategy for optimizing non-linear least-squares problems.

$$\arg\min_{\mathbf{y}} ||\mathbf{x} - \mathcal{F}(\mathbf{y})||_2^2$$
$$\text{s.t. } \mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

"Carl Friedrich Gauss"

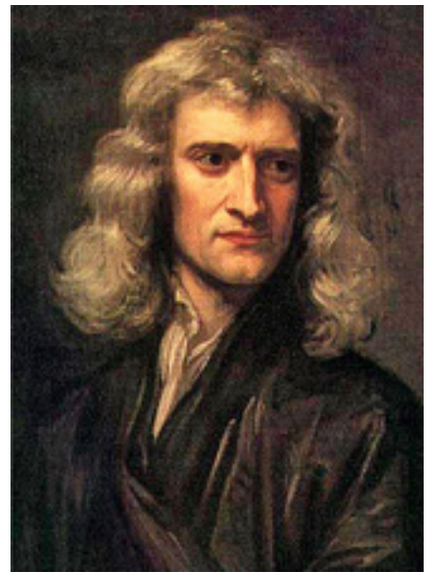
Step 1:

$$\arg\min_{\Delta\mathbf{y}} ||\mathbf{x} - \mathcal{F}(\mathbf{y}) - \frac{\partial \mathcal{F}(\mathbf{y})}{\partial \mathbf{y}^T}\Delta\mathbf{y}||_2^2$$

Step 2:

$$\mathbf{y} \rightarrow \mathbf{y} + \Delta\mathbf{y}$$

keep applying steps until $\Delta\mathbf{y}$ converges.



"Isaac Newton"

Reminder: Gauss-Newton Algorithm

- Gauss-Newton (GN) algorithm common strategy for optimizing non-linear least-squares problems.

$$\arg\min_{\mathbf{y}} ||\mathbf{x} - \mathcal{F}(\mathbf{y})||_2^2$$
$$\text{s.t. } \mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

"Carl Friedrich Gauss"

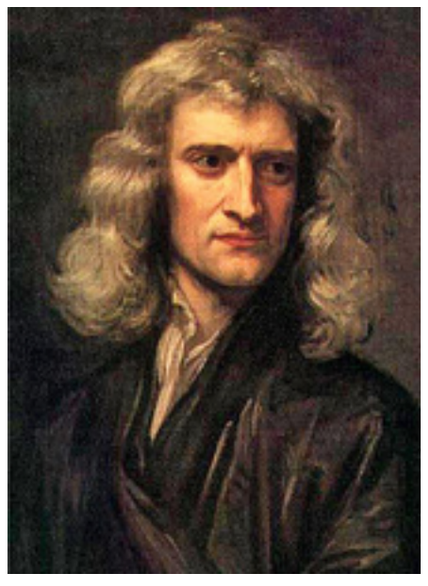
Step 1:

$$\arg\min_{\Delta\mathbf{y}} ||\mathbf{x} - \mathcal{F}(\mathbf{y}) - \frac{\partial \mathcal{F}(\mathbf{y})}{\partial \mathbf{y}^T}\Delta\mathbf{y}||_2^2$$

Step 2:

$\mathbf{y} \rightarrow \mathbf{y} + \Delta\mathbf{y}$ **"Is the update additive?"**

keep applying steps until $\Delta\mathbf{y}$ converges.



"Isaac Newton"

Today

- SfM - Bundle Adjustment
- VSLAM - Keyframe vs. Filtering
- Visual Odometry
- Loop Closure

Mono SLAM = Online SFM

- Monocular SLAM is just another name for “online” SFM.
- If computation was not an issue, one would just apply Bundle Adjustment after every new frame

$$\arg\min_{\mathbf{w},\boldsymbol{\theta}} \sum_{f=1}^{F} \sum_{n=1}^{N} ||\mathbf{x}_n^f - \pi(\mathbf{w}_n; \boldsymbol{\theta}^f)||_2^2$$

$\mathbf{x} \leftarrow$ 2D projection $\mathbf{w} \leftarrow$ 3D point

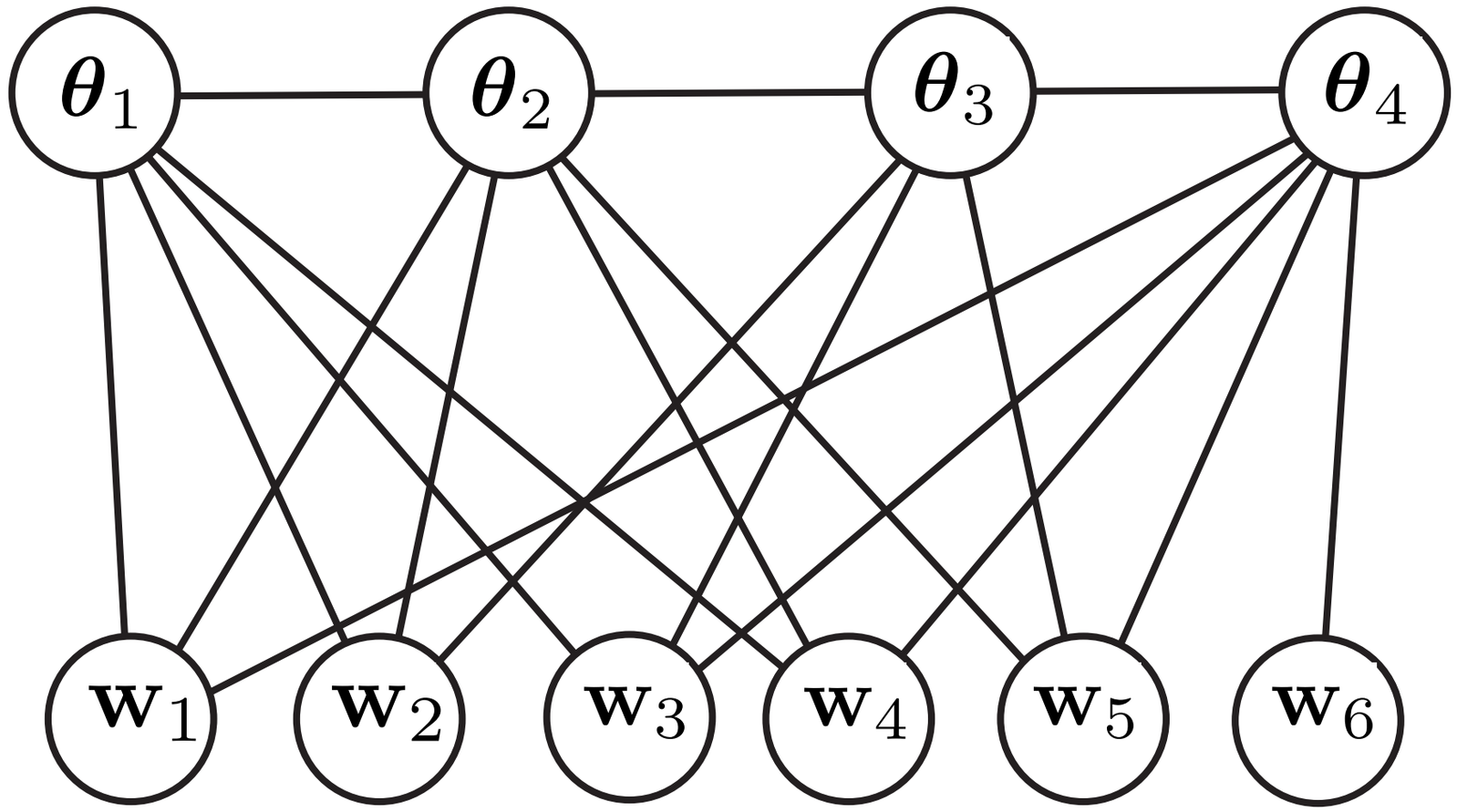
$\boldsymbol{\theta} \leftarrow$ extrinsics $N \leftarrow$ no. of points

$\pi \leftarrow$ projection function

$F \leftarrow$ no. of frames

Mono SLAM - MRF

- One can view the problem of SfM - Bundle Adjustment as doing inference on a Markov Random Field (MRF).
- **Problem -** becomes exponentially harder as times goes on.

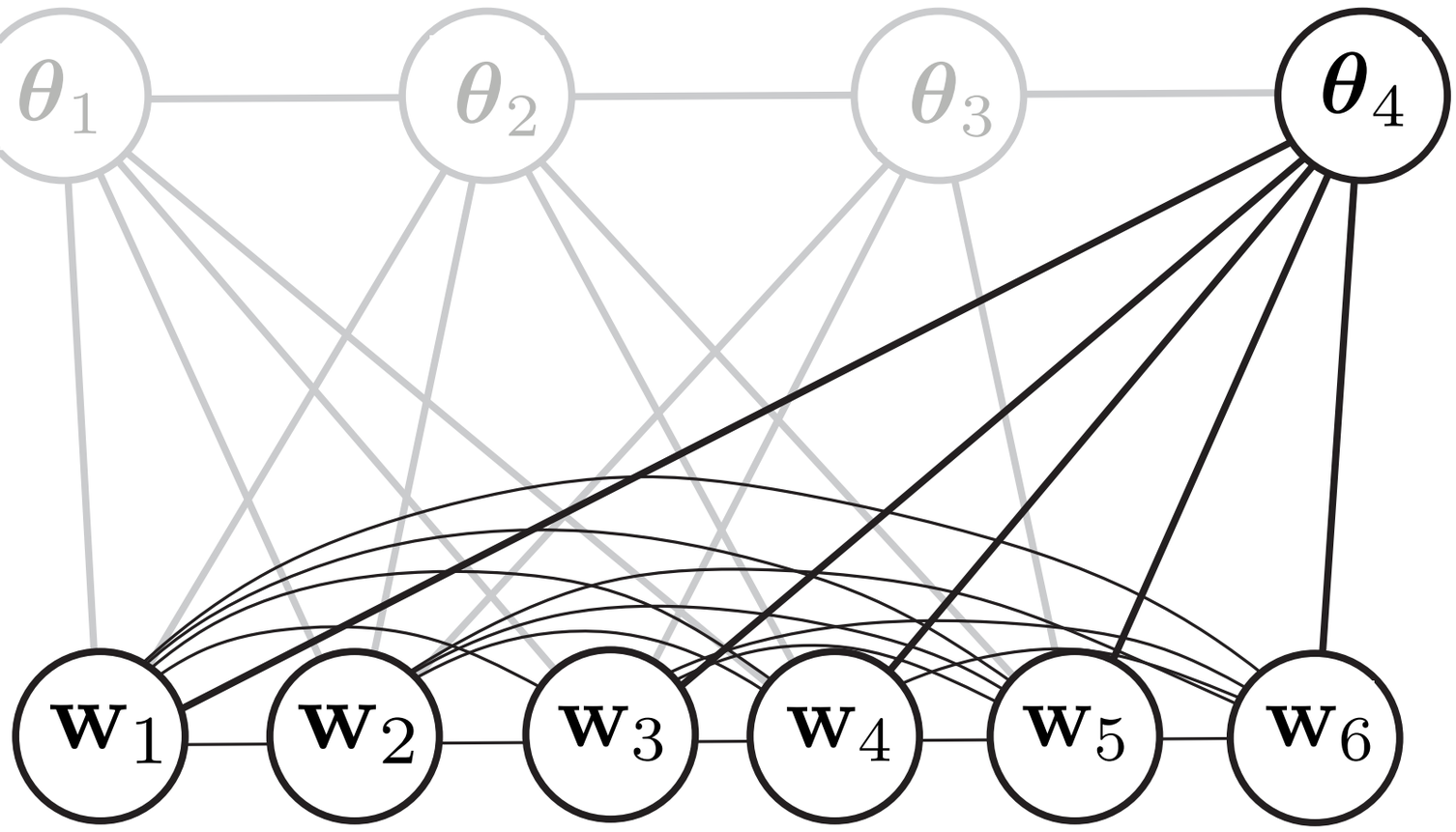


"edges based on visibility" ρ

H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" Image and Vision Computing, vol. 30, no. 2, pp. 65–77, 2012.

Mono SLAM - Filtering

- Classic way of resolving this was to pose BA problem as a filter - such as an Extended Kalman Filter (EKF).
- **Problem** - Wastes processing time on frames that added very little information.

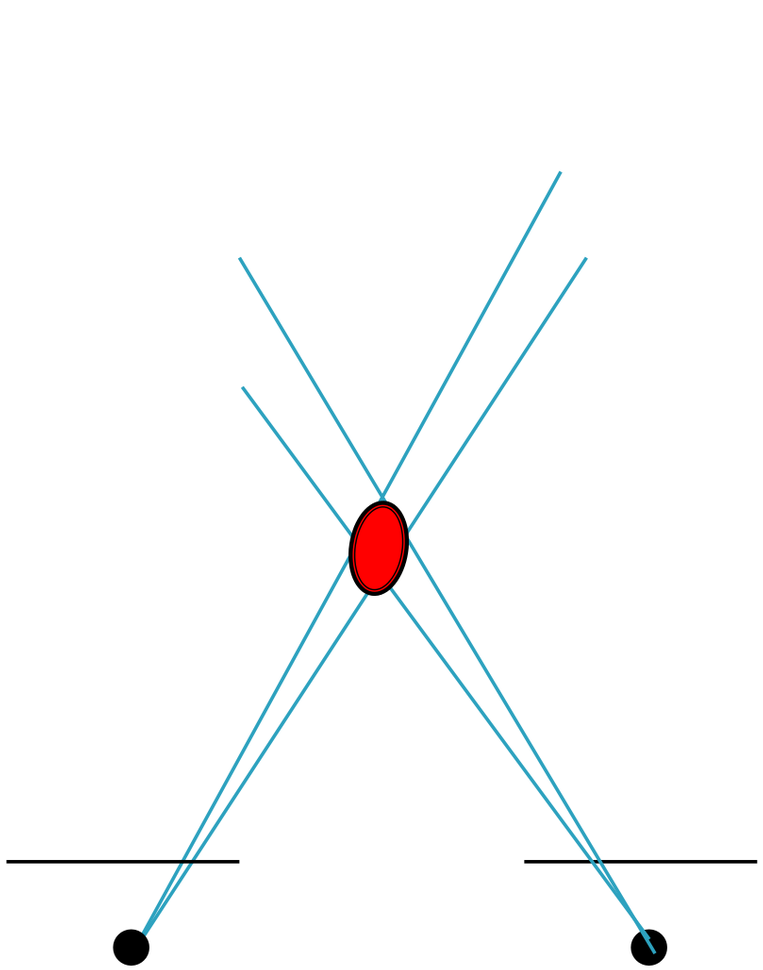


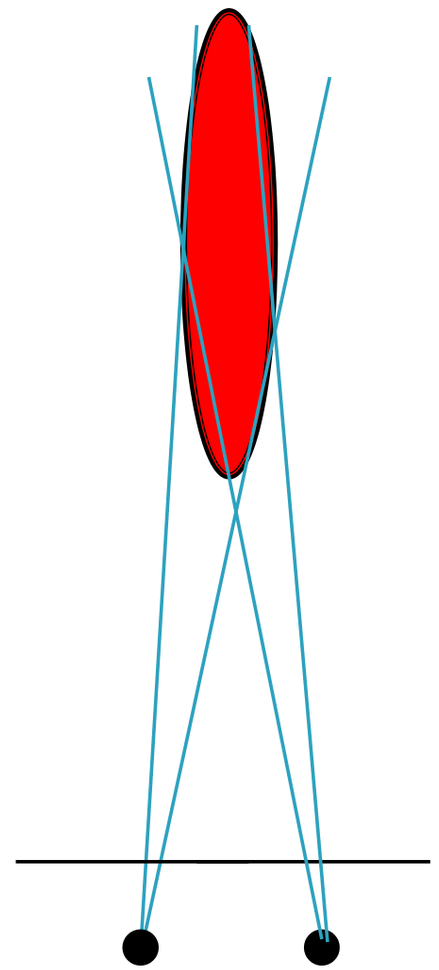
"marginalizing out previous poses also results in **unwanted** direct connections between 3D points"

H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" Image and Vision Computing, vol. 30, no. 2, pp. 65–77, 2012.

Mono SLAM - Filtering

- Filtering approaches are often times problematic (e.g. think when the device stops moving).
- When frames are taken at nearby positions compared to the scene distance, 3D points will exhibit large uncertainty.

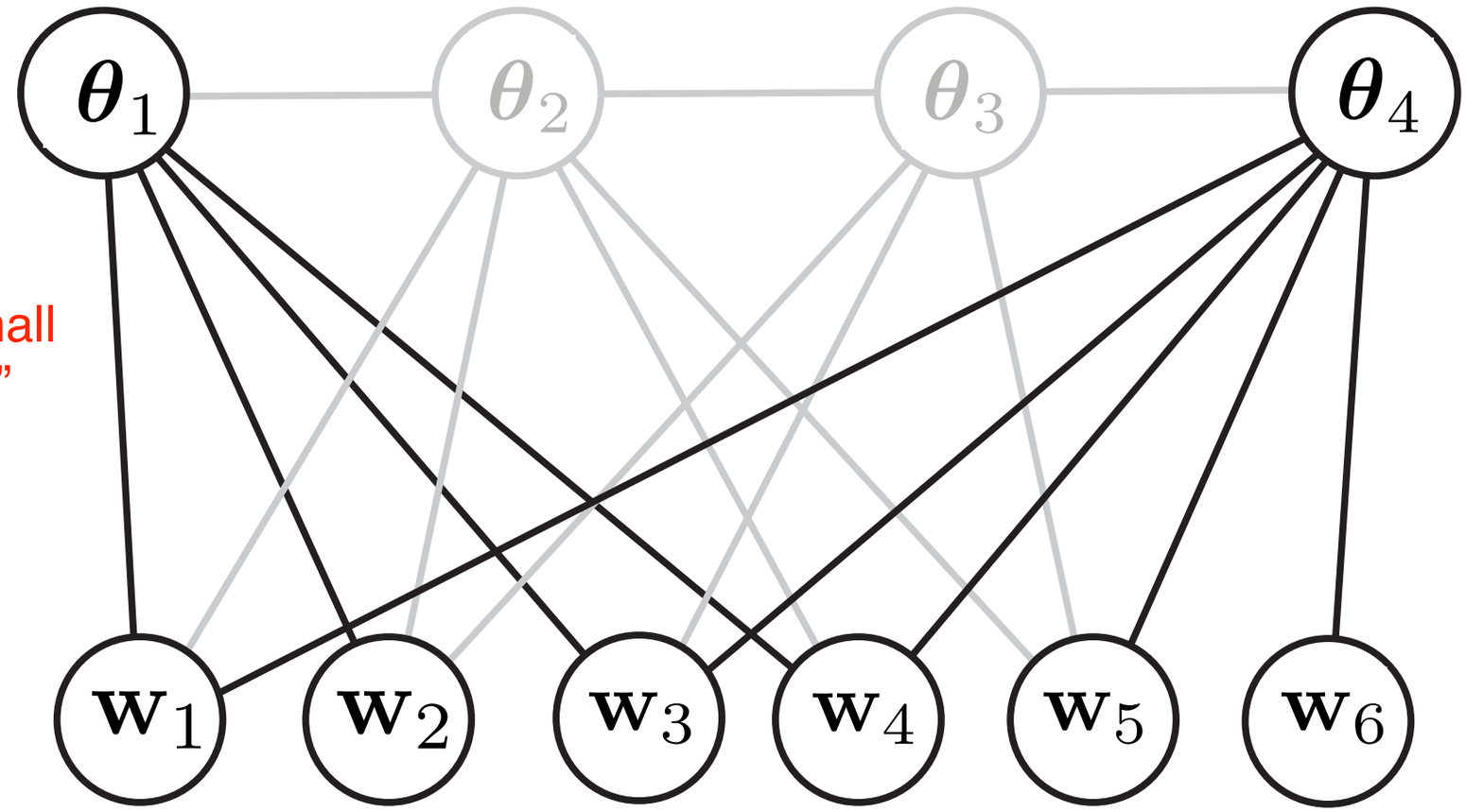




Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

Mono SLAM - Keyframe

- A better strategy is to employ **keyframe BA**.
- Made popular by Klein & Murray's - Parallel Tracking and Mapping (PTAM) algorithm.



"remove **all** but a small subset of keyframes"

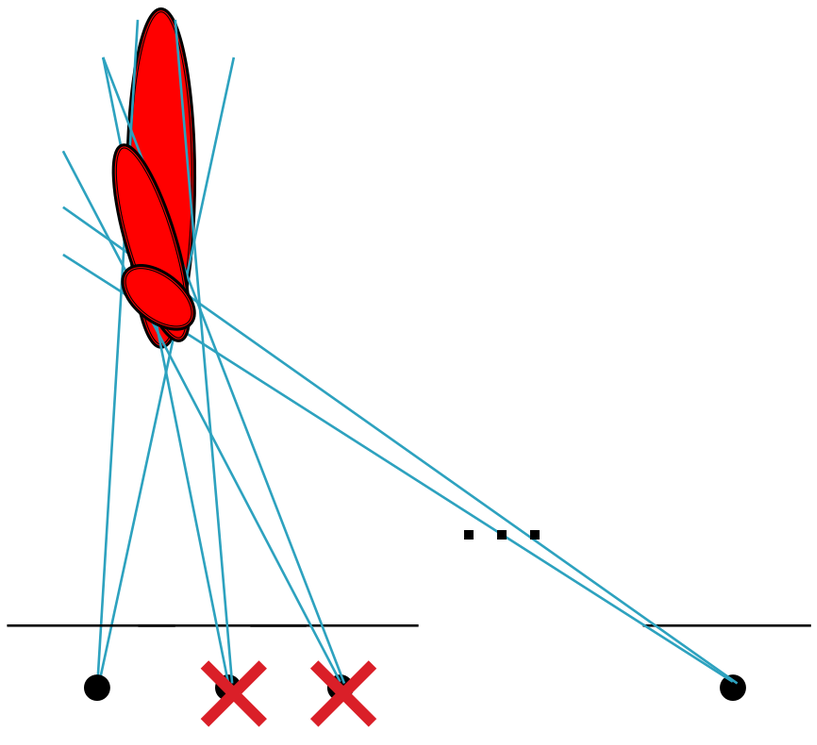
G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces", ISMAR 2007.

H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" Image and Vision Computing, vol. 30, no. 2, pp. 65–77, 2012.

Keyframe Selection

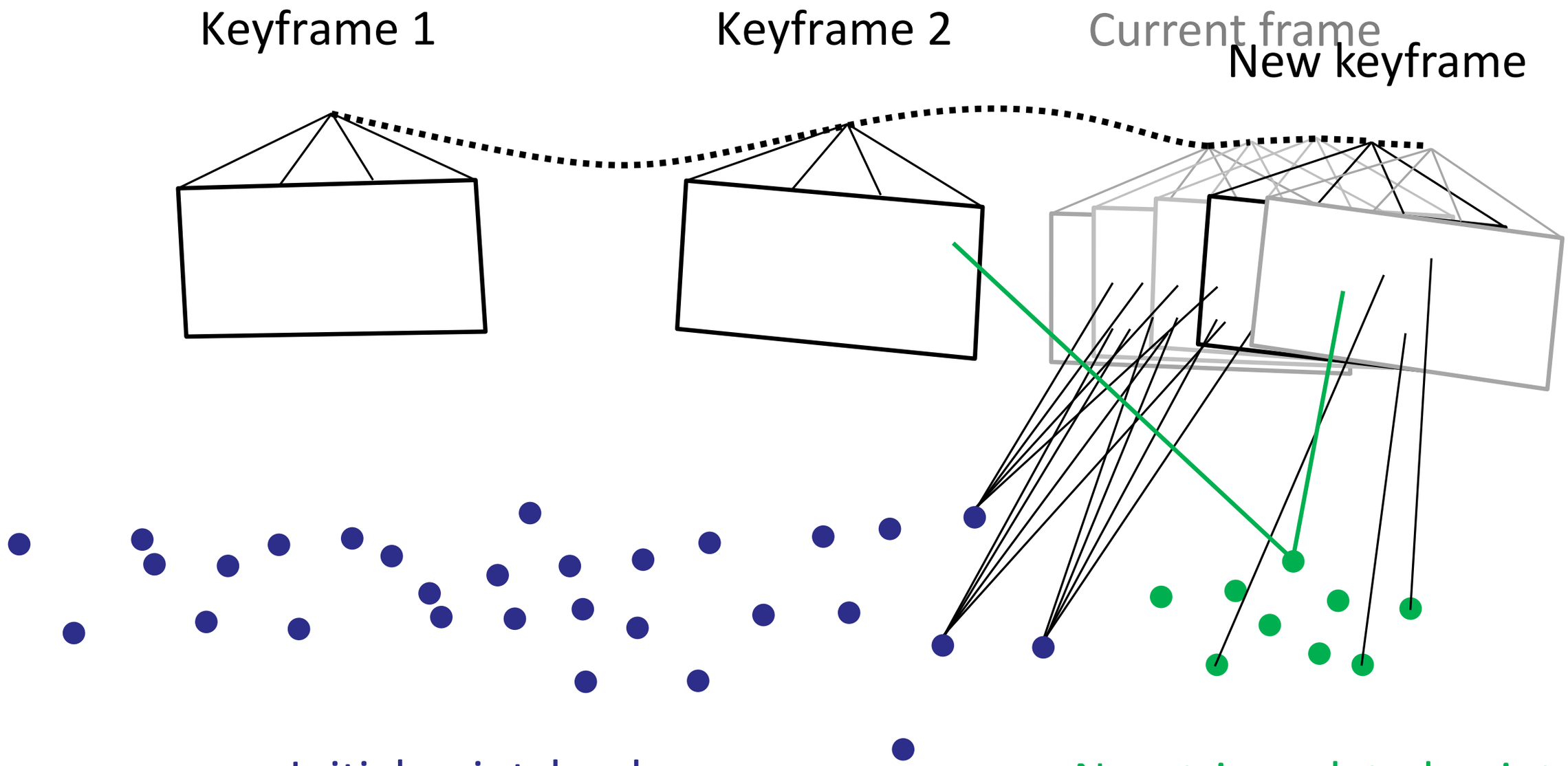
- One way to avoid this consists of skipping frames until the average uncertainty of the 3D points decreases below a certain threshold. The selected frames are called **keyframes**.
- Rule of thumb: add a keyframe when,

$$\frac{keyframe\ distance}{average\text{-}depth} > threshold\ (\sim 10\text{-}20\ \%)$$



Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

Keyframe-based SLAM



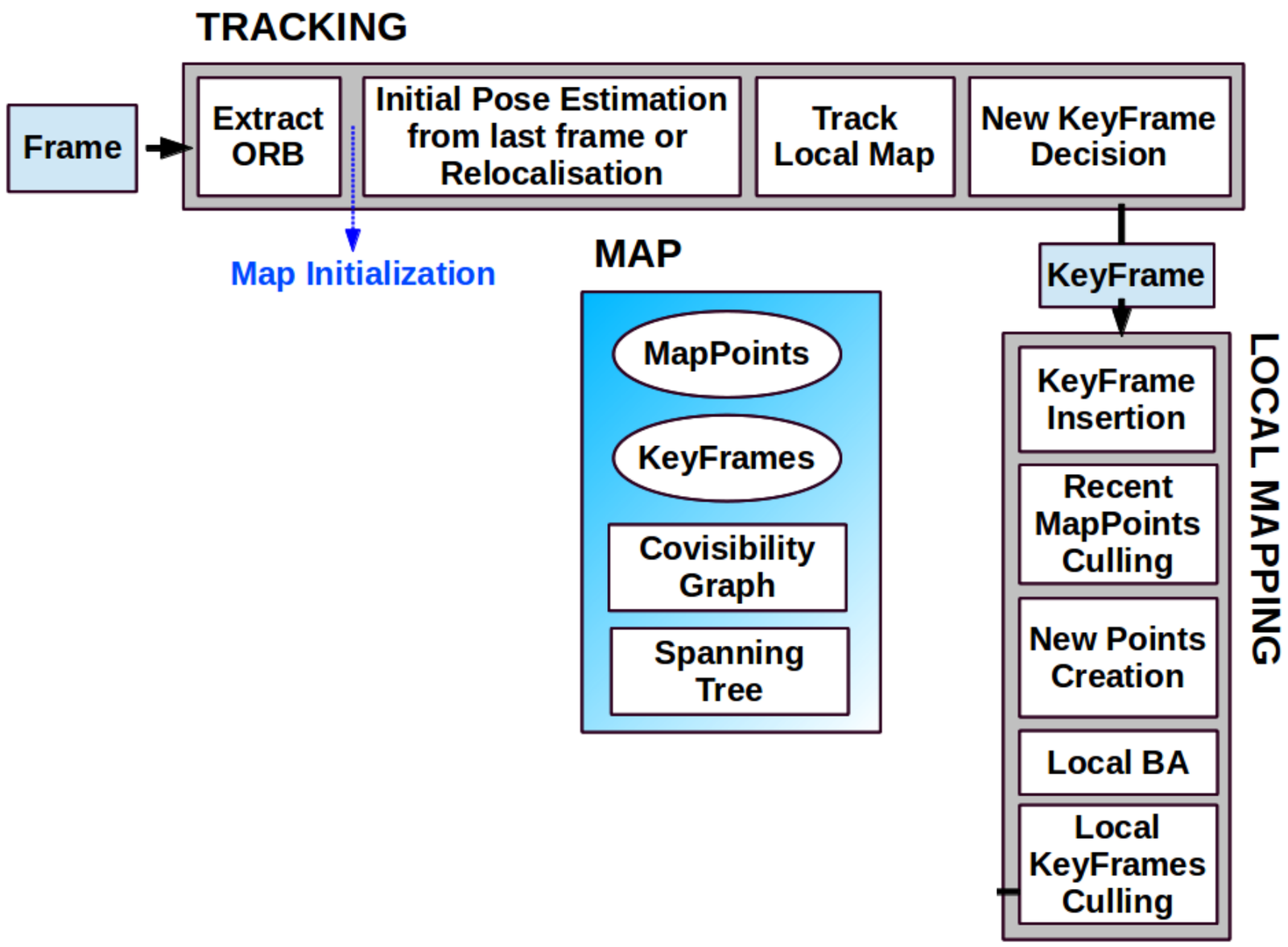
Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

PTAM - Separate Threads

- An innovation of Klein & Murray's PTAM method was the separation of the camera tracking (θ) and map estimation (w) tasks.
- Camera Tracking or **visual odometery** (**VO**) runs on one thread in real-time.
- Map estimation runs on a separate thread (not having to run in real-time) allowing for bundle adjustment.
- Same idea is still being utilized in current state of the art visual SLAM algorithms (e.g. ORB SLAM).

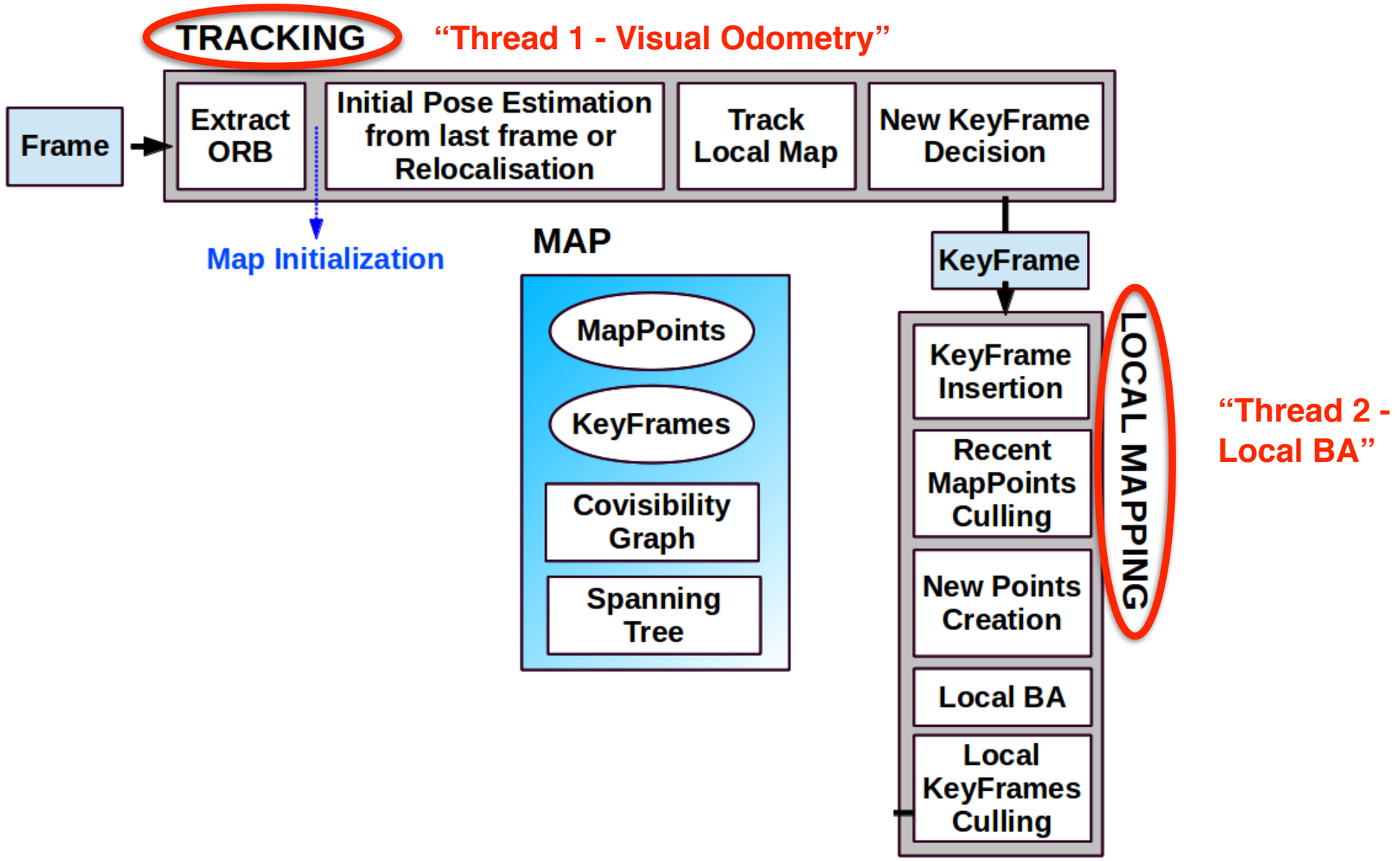
G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces", ISMAR 2007.

Example - ORB SLAM



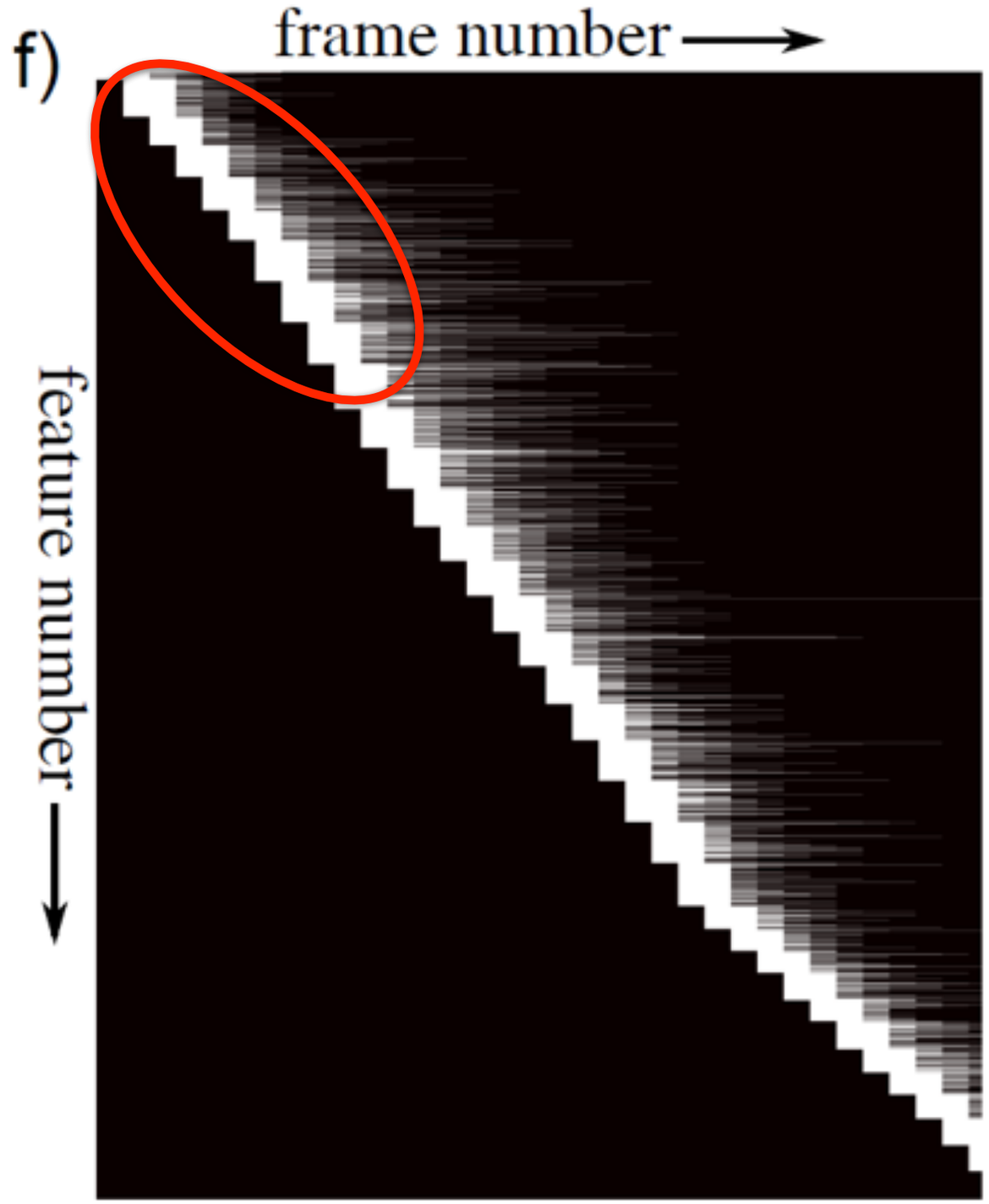
R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, “ORB-SLAM: a Versatile and Accurate Monocular SLAM System” IEEE Trans. Robotics 2015.

Example - ORB SLAM



R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System" IEEE Trans. Robotics 2015.

Local Bundle Adjustment

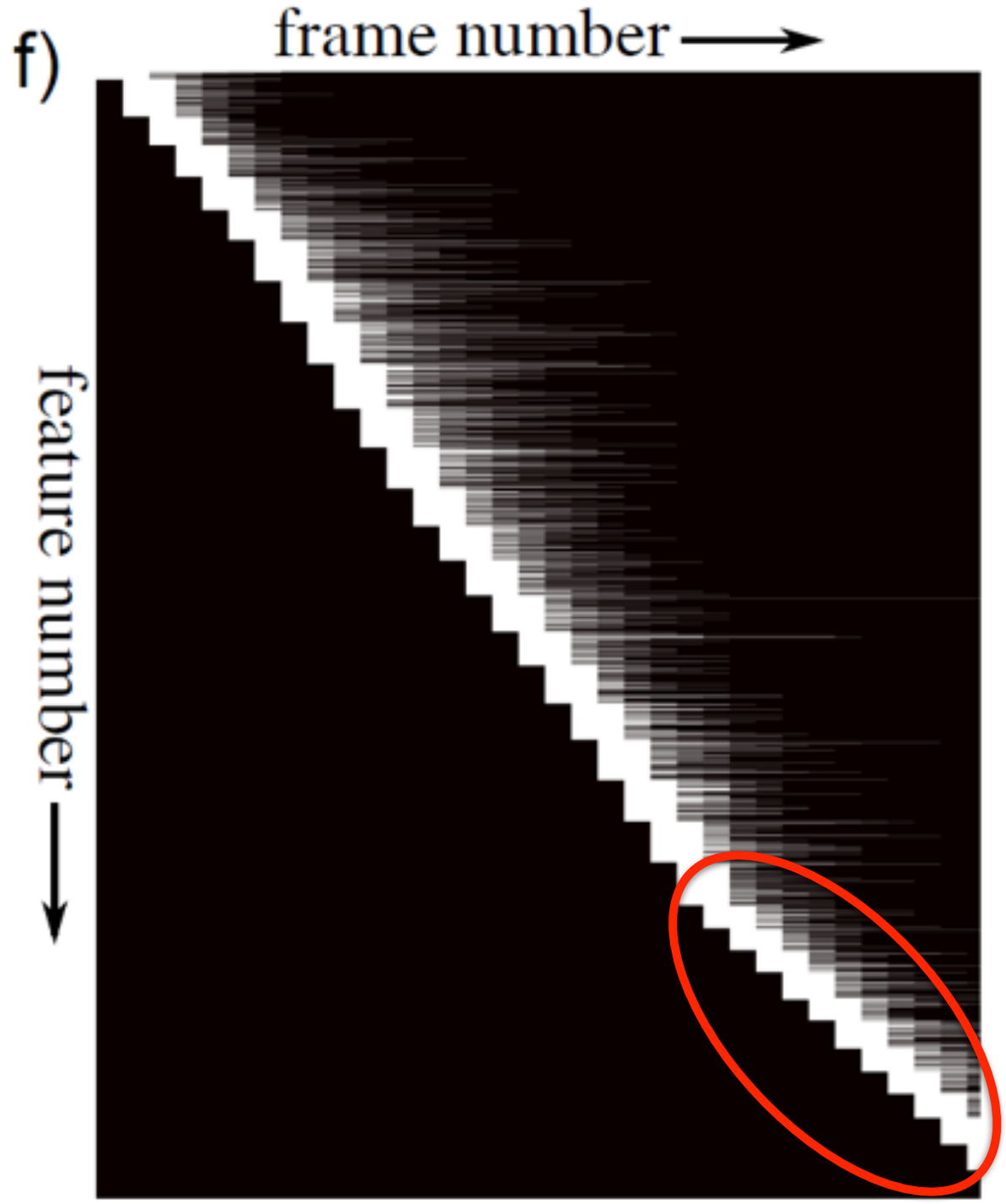


$$\Upsilon = \begin{bmatrix} \rho_1^1 & \cdots & \rho_1^F \\ \vdots & \ddots & \vdots \\ \rho_N^1 & \cdots & \rho_N^F \end{bmatrix}$$

"visibility matrix"

M. Kaess, A. Ranganathan & F. Dellaert, "iSAM: Incremental Smoothing and Mapping" IEEE Trans. Robotics 2008.

Local Bundle Adjustment



$$\Upsilon = \begin{bmatrix} \rho_1^1 & \cdots & \rho_1^F \\ \vdots & \ddots & \vdots \\ \rho_N^1 & \cdots & \rho_N^F \end{bmatrix}$$

"visibility matrix"

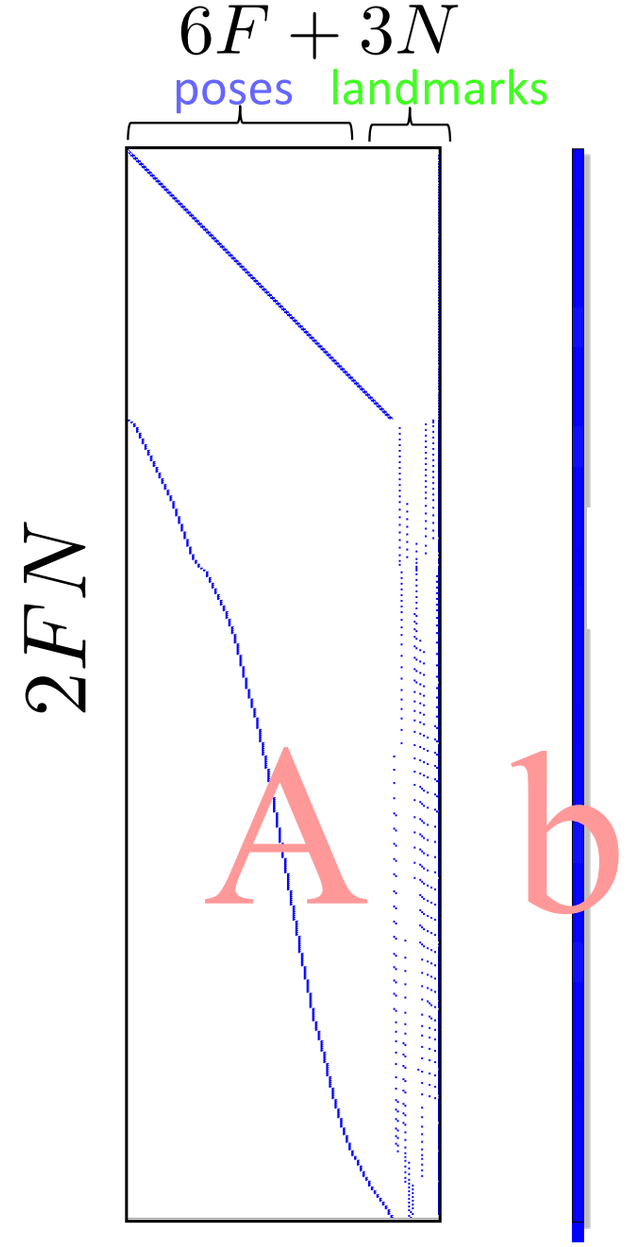
M. Kaess, A. Ranganathan & F. Dellaert, "iSAM: Incremental Smoothing and Mapping" IEEE Trans. Robotics 2008.

iSAM: Incremental Smoothing & Mapping

$$\arg \min_{\Delta\boldsymbol{\theta},\Delta\mathbf{w}} ||\mathbf{b} - \mathbf{A} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{w} \end{bmatrix} ||_2^2$$

- iSAM can be entertained using QR Factorization.

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

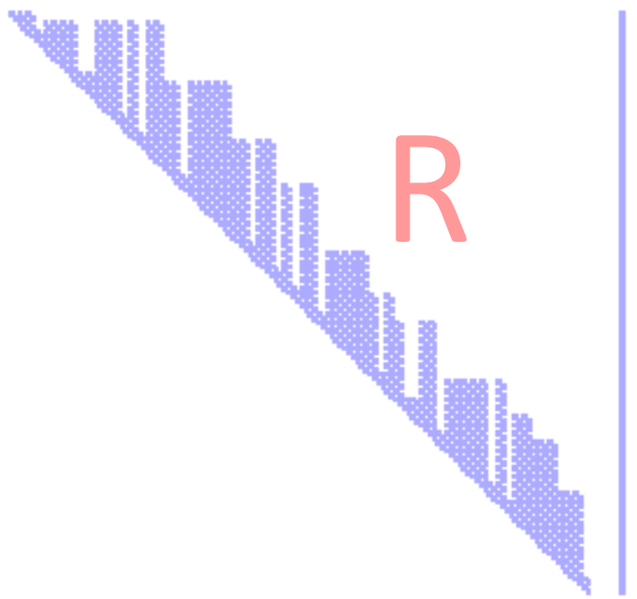


M. Kaess, A. Ranganathan & F. Dellaert, “iSAM: Incremental Smoothing and Mapping” IEEE Trans. Robotics 2008.

iSAM: Incremental Smoothing & Mapping

Solving a growing system:

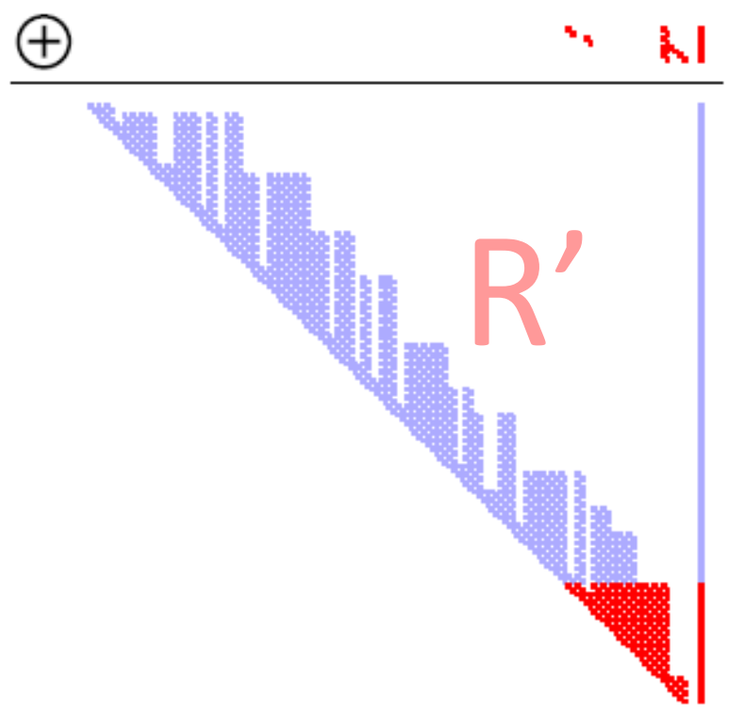
- R factor from previous step
- How do we add new measurements?

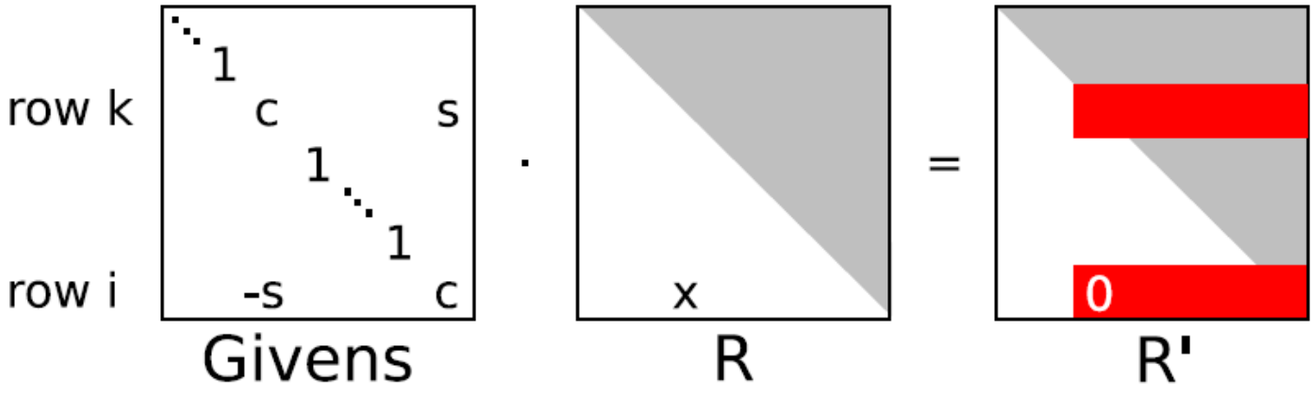


Key idea:

- Append to existing matrix factorization
- "Repair" using Givens rotations

New measurements ->





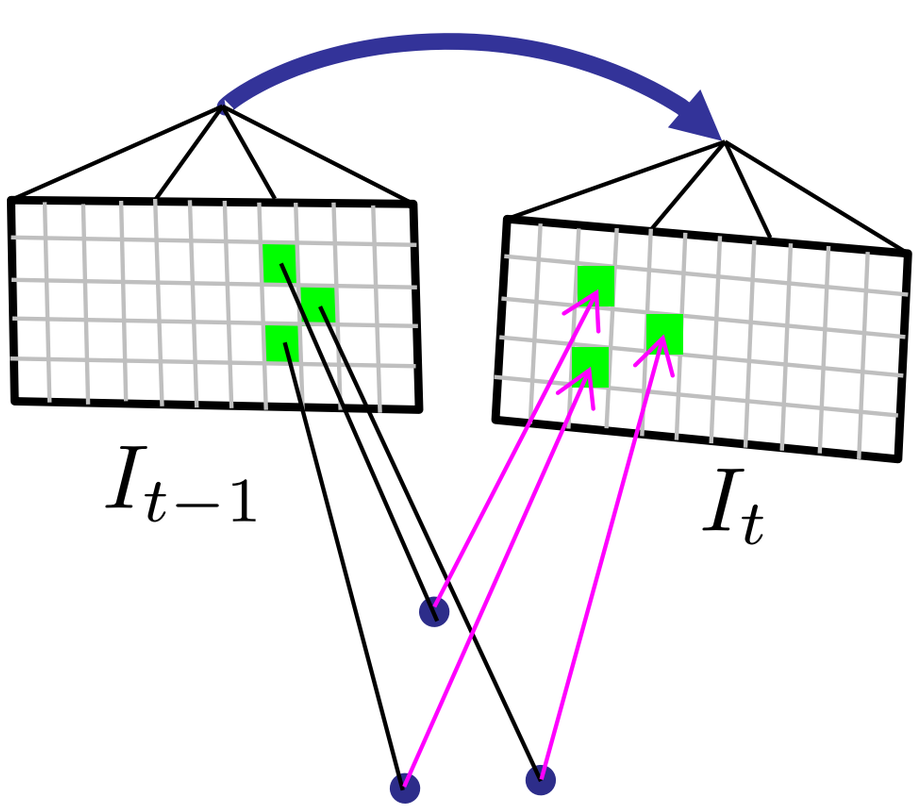
M. Kaess, A. Ranganathan & F. Dellaert, "iSAM: Incremental Smoothing and Mapping" IEEE Trans. Robotics 2008.

Today

- SfM - Bundle Adjustment
- VSLAM - Keyframe vs. Filtering
- Visual Odometry
- Loop Closure

VO vs SFM

- VO is a **particular case** of SFM
- VO focuses on estimating the 3D motion of the camera **sequentially** (as a new frame arrives) and in **real time**.
- Terminology: sometimes SFM is used as a synonym of VO.



Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

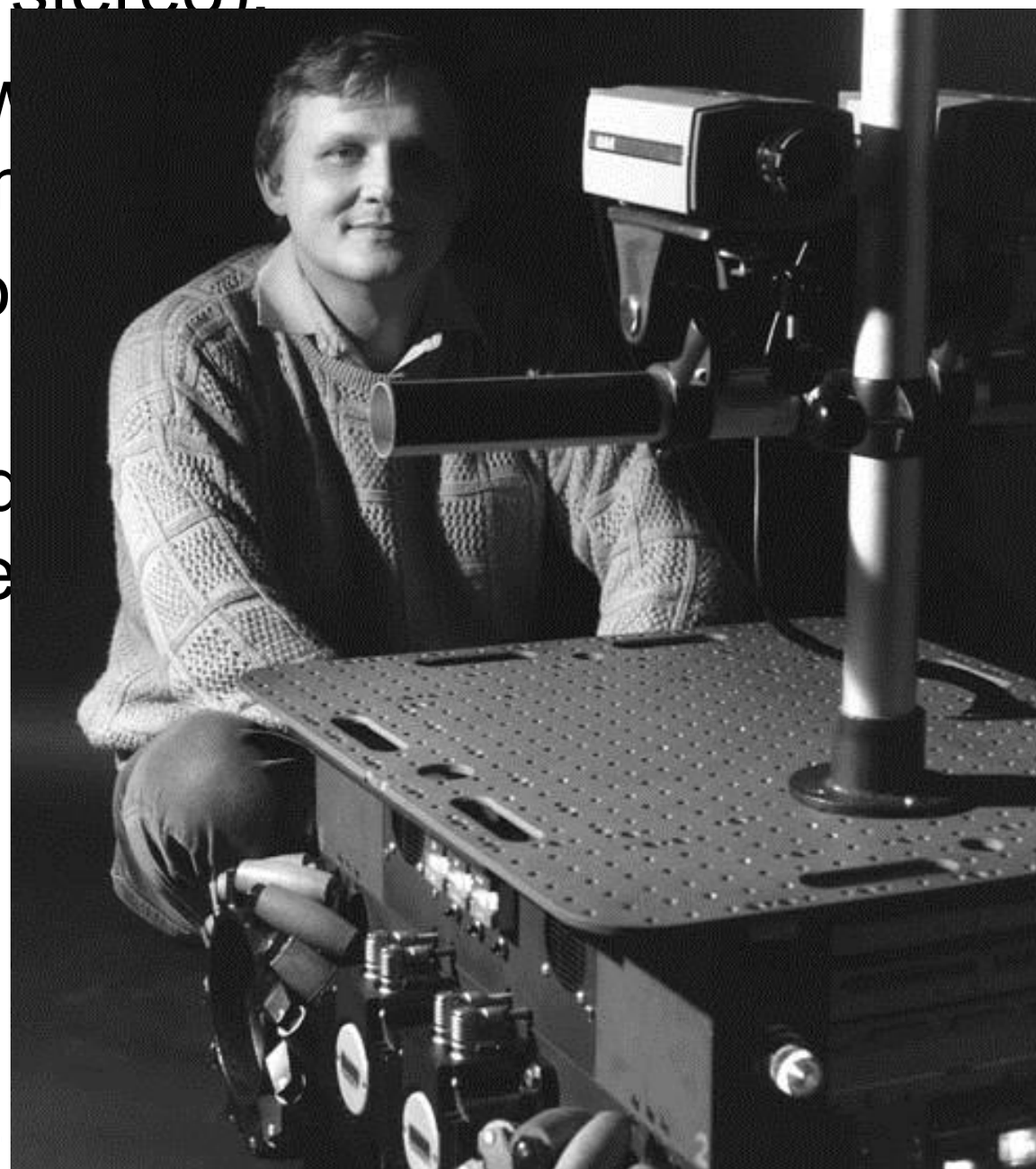
A Brief History of VO

- **1980**: First known VO real-time implementation on a robot by Hans Moraveck PhD thesis (NASA/JPL) for Mars rovers using one sliding camera (sliding stereo).
- **1980 to 2000**: The VO research was dominated by NASA/ JPL in preparation of 2004 Mars mission.
- **2004**: VO used on a robot on another planet: Mars rovers Spirit and Opportunity
- **2004**. VO was revived in the academic environment by Nister et al. The term VO became popular.

Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

A Brief History of VO

- **1980**: First known VO real-time implementation on a robot by Hans Moraveck PhD thesis (NASA/JPL) for Mars rovers using one sliding camera (sliding stereo).
- **1980 to 2000**: The VO research w
  JPL in preparation of 2004 Mars r
- **2004**: VO used on a robot on ano
  Spirit and Opportunity
- **2004**. VO was revived in the acad
  Nister et al. The term VO became

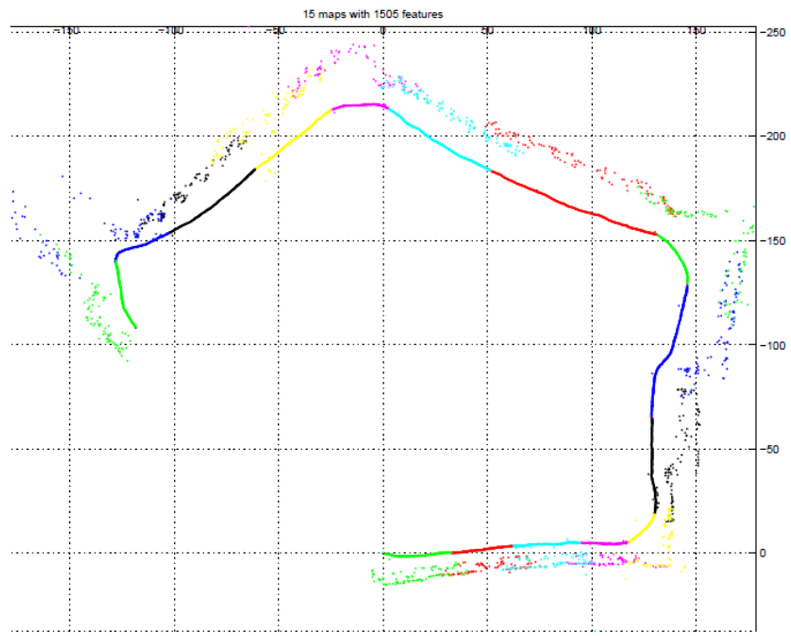


Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

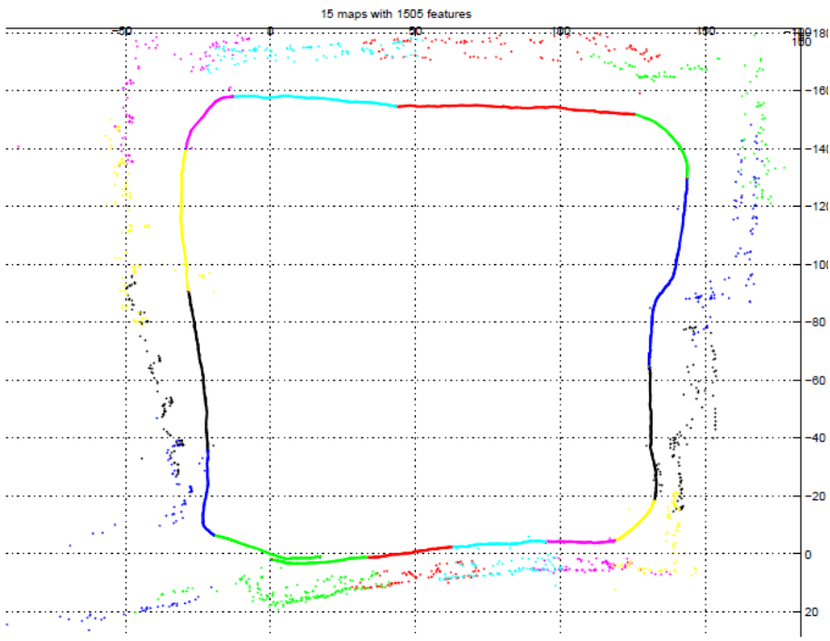
VO vs Visual SLAM

- VO only aims to the local consistency of the trajectory.
- SLAM aims to the global consistency of the trajectory and of the map.
- VO can be used as a building block of SLAM.
- VO is SLAM before **closing the loop!**
- The choice between VO and V-SLAM depends on the tradeoff between performance and consistency, and simplicity in implementation.
- VO trades off consistency for real-time performance, without the need to keep track of all the previous history of the camera.

Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

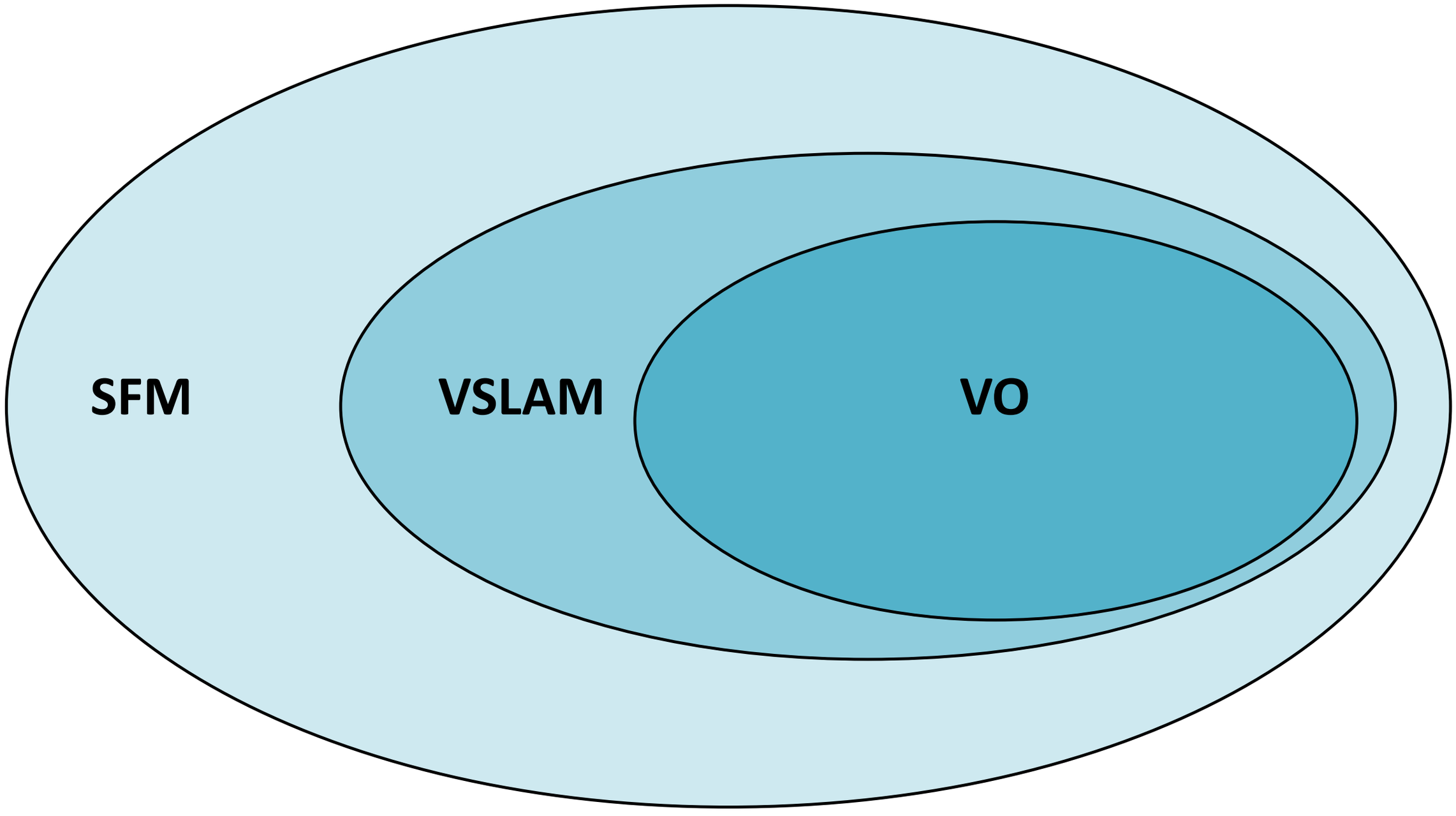


Visual odometry



Visual SLAM

VO vs VSLAM vs SFM



Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

Today

- SfM - Bundle Adjustment
- VSLAM - Keyframe vs. Filtering
- Visual Odometry
- Loop Closure

Loop Closure Detection

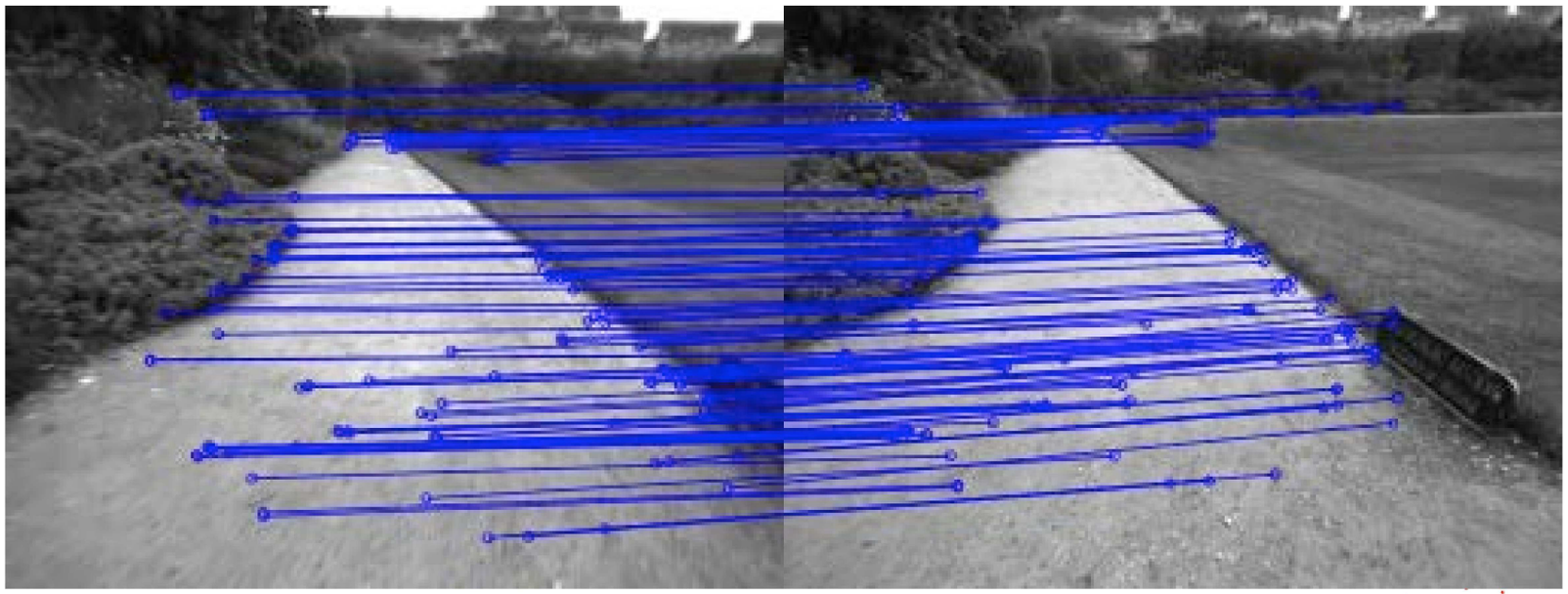
- Loop constraints are very valuable constraints for local BA.
- Loop constraints can be found by evaluating visual similarity between the current camera images and past camera images.
- Visual similarity can be computed using global image descriptors (GIST descriptors) or local image descriptors (e.g., ORB features).
- Image retrieval is the problem of finding the most similar image of a template image in a database of billion images (image retrieval).
- This can be solved efficiently with Bag of Words.

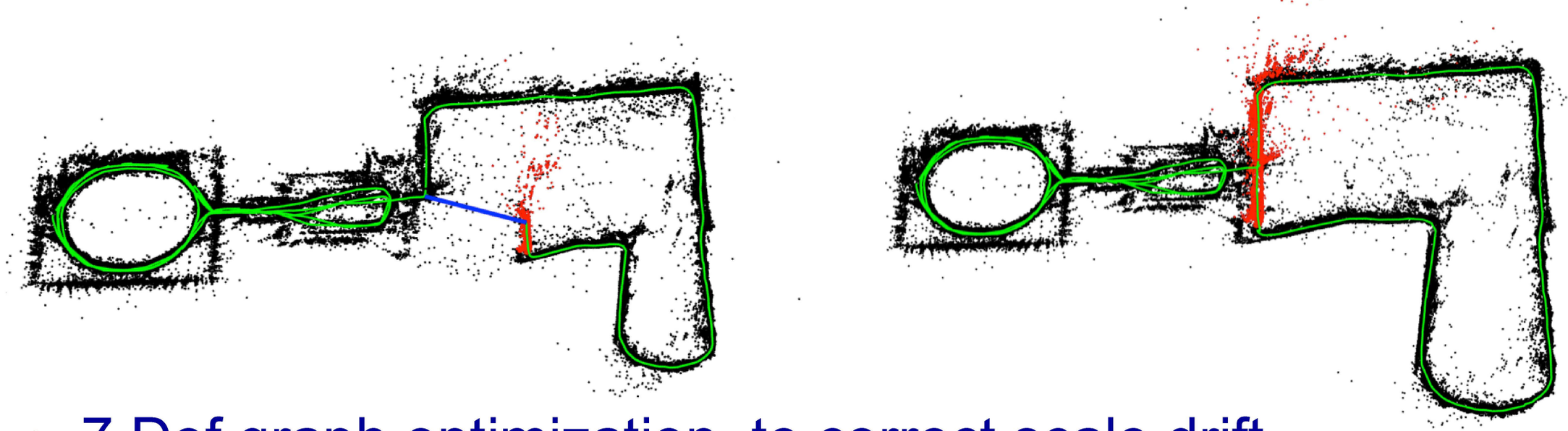
First observation

Second observation after a loop

Taken from D. Scaramuzza “Tutorial on Visual Odometry”.

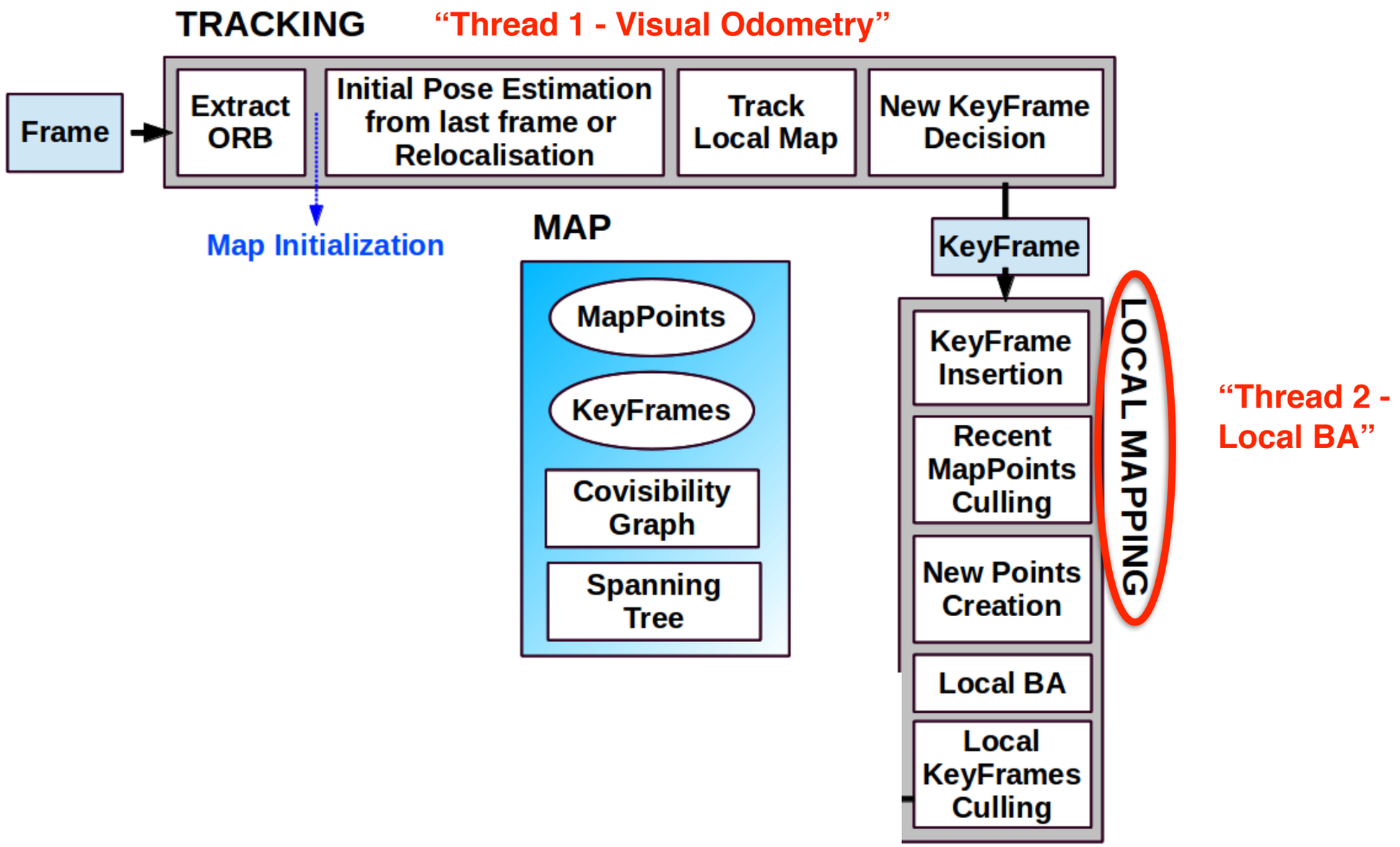
Loop Closure Detection





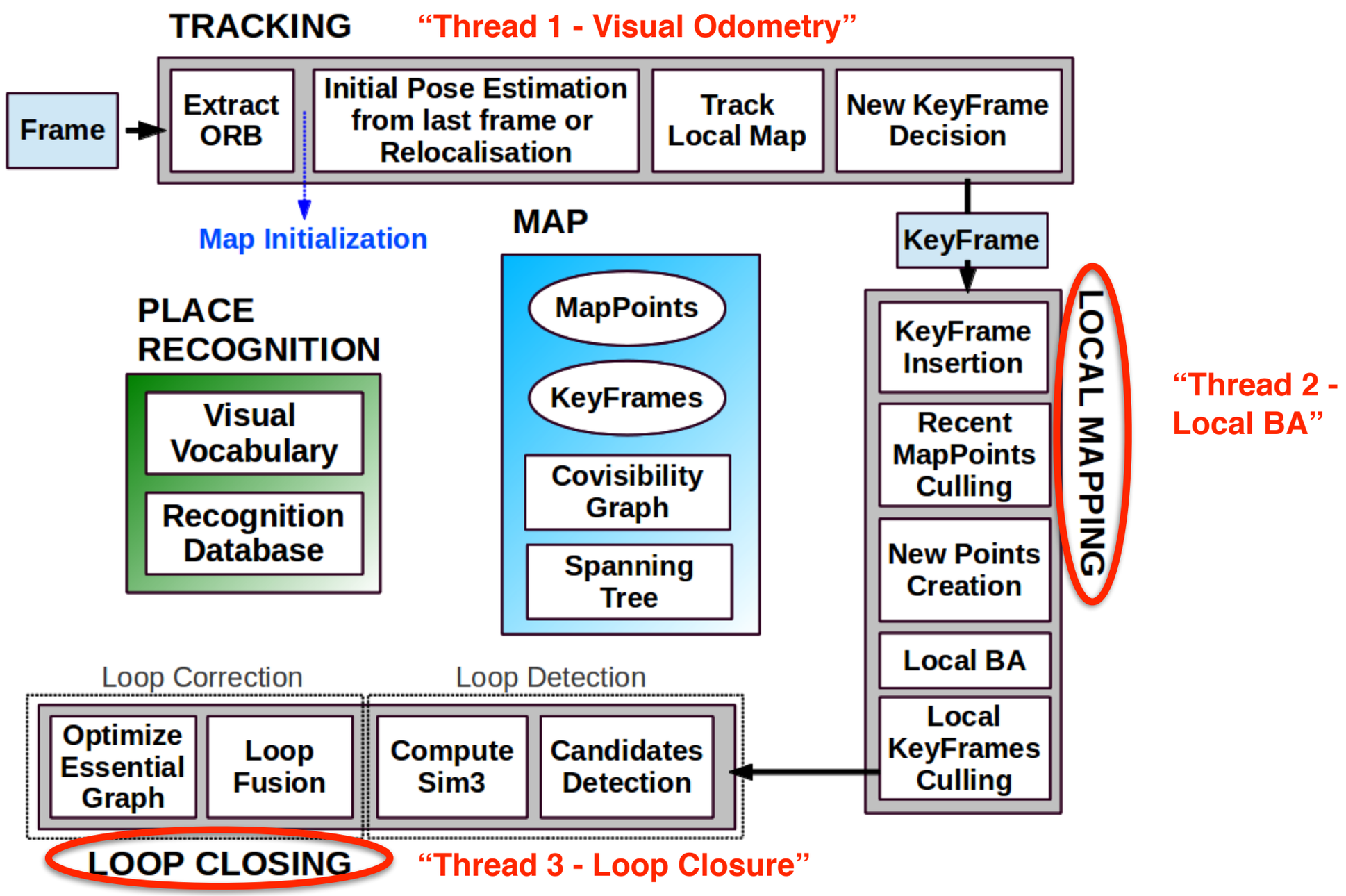
- 7 Dof graph optimization, to correct scale drift
- And optionally Full BA (little improvement, much slower)

Example - ORB SLAM



R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System" IEEE Trans. Robotics 2015.

Example - ORB SLAM



R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System" IEEE Trans. Robotics 2015.

ORB SLAM

- Essentially “greatest hits” in terms of work in VSLAM.
- Uses the same features (i.e. ORB) for,
  - Tracking
  - Mapping
  - Loop closure
- Real-time, large scale operation.
- Survival of the fittest for points and keyframes.
- Further information can be found at,
  - http://webdiis.unizar.es/~raulmur/orbslam/
  - Source code available under GPLv3.

R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, “ORB-SLAM: a Versatile and Accurate Monocular SLAM System” IEEE Trans. Robotics 2015.

ORB SLAM

Taken from: Mur-Artal, Raul, J. M. M. Montiel, and Juan D. Tardós. "Orb-slam: a versatile and accurate monocular slam system." IEEE Transactions on Robotics 31.5 (2015): 1147-1163.

ORB SLAM

Taken from: Mur-Artal, Raul, J. M. M. Montiel, and Juan D. Tardós. "Orb-slam: a versatile and accurate monocular slam system." IEEE Transactions on Robotics 31.5 (2015): 1147-1163.