# Robotic 3D Vision

## Lecture 10: Visual SLAM 1 – Introduction, Bundle Adjustment

WS 2020/21

Dr. Niclas Zeller
Artisense GmbH

# What We Will Cover Today

- Introduction to Visual SLAM
- Formulation of the SLAM Problem
- Full SLAM Posterior
- Bundle Adjustment (BA)
- Structure of the SLAM/BA Problem

# What is Visual SLAM?

- Visual simultaneous localization and mapping (VSLAM)…
  - Tracks the pose of the camera in a map, and simultaneously
  - Estimates the parameters of the environment map (f.e. reconstruct the 3D positions of interest points in a common coordinate frame)
- Loop-closure: Revisiting a place allows for drift compensation
  - How to detect a loop closure

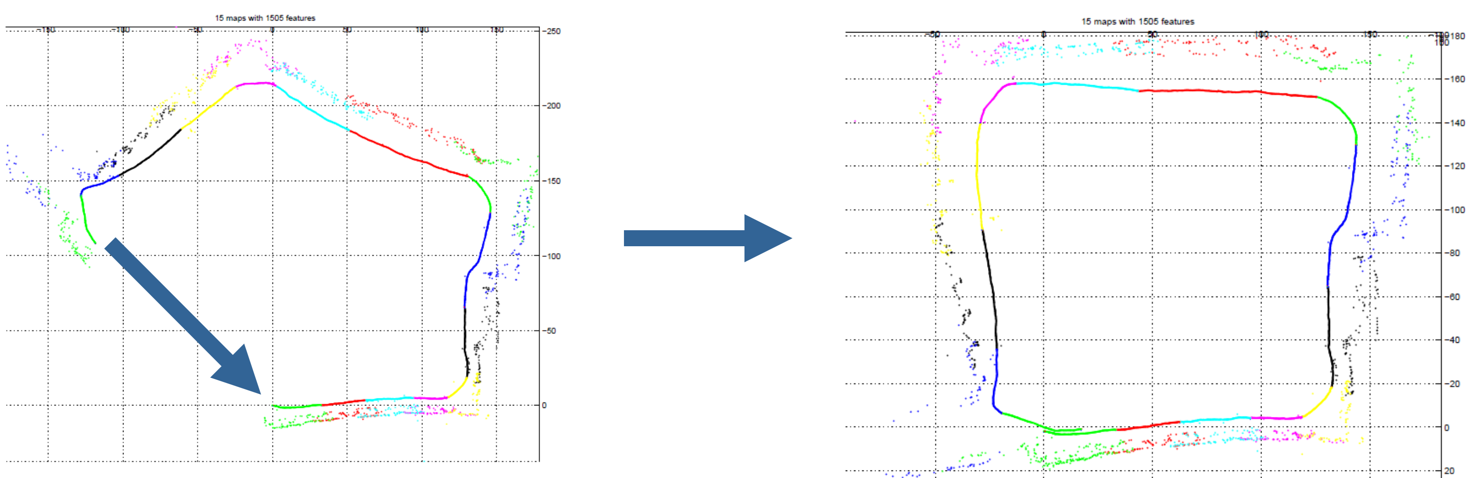
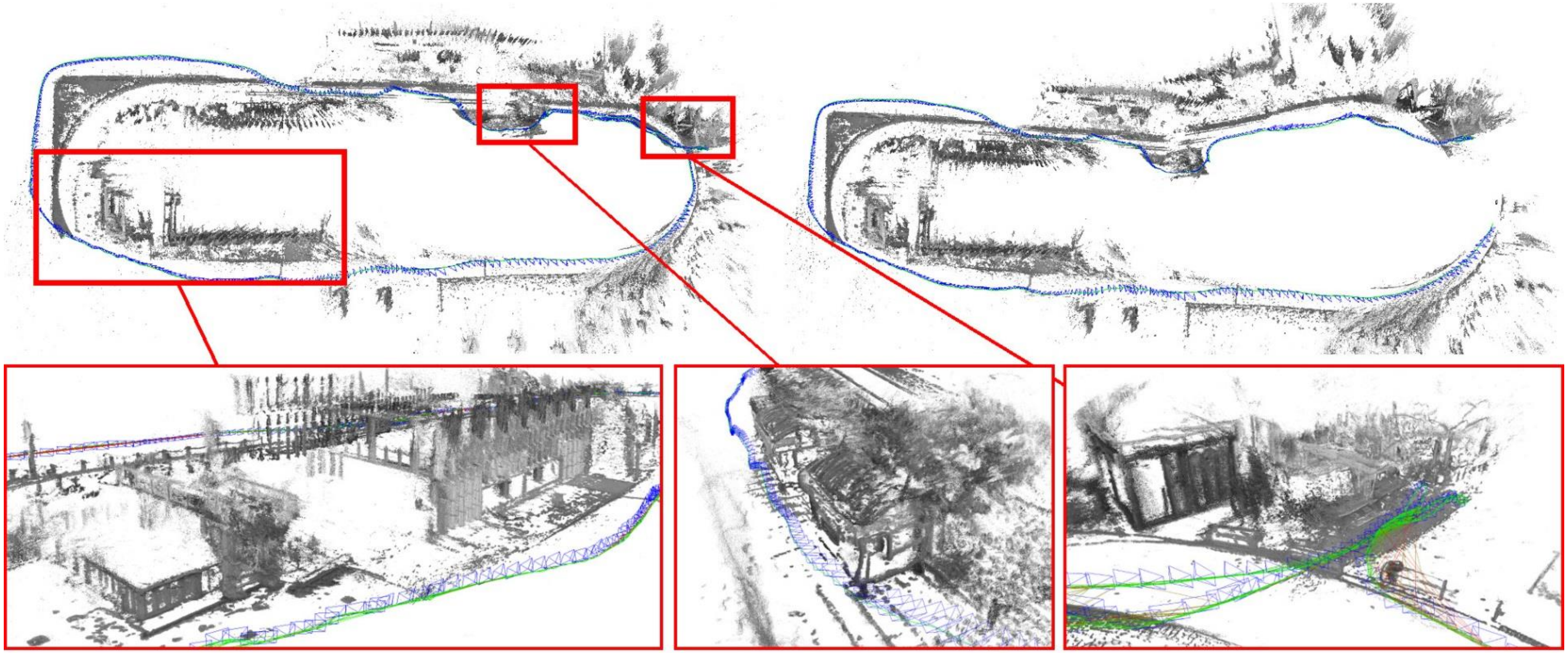



Image credit: Clemente et al., RSS 2007

# What is Visual SLAM?

- Visual simultaneous localization and mapping (VSLAM)…
  - Tracks the pose of the camera in a map, and simultaneously
  - Estimates the parameters of the environment map (f.e. reconstruct the 3D positions of interest points in a common coordinate frame)
- Loop-closure: Revisiting a place allows for drift compensation
  - How to detect a loop closure
- Global vs. local optimization methods
  - Global: bundle adjustment, pose-graph optimization, etc.
  - Local: incremental tracking-and-mapping approaches, visual odometry with local maps. Designed for real-time operation.
  - Hybrids: Real-time local tracking and mapping + global optimization in a slower parallel process (LSD-SLAM, ORB-SLAM, LDSO, etc.)

# Visual SLAM using Pose Graph Opt.



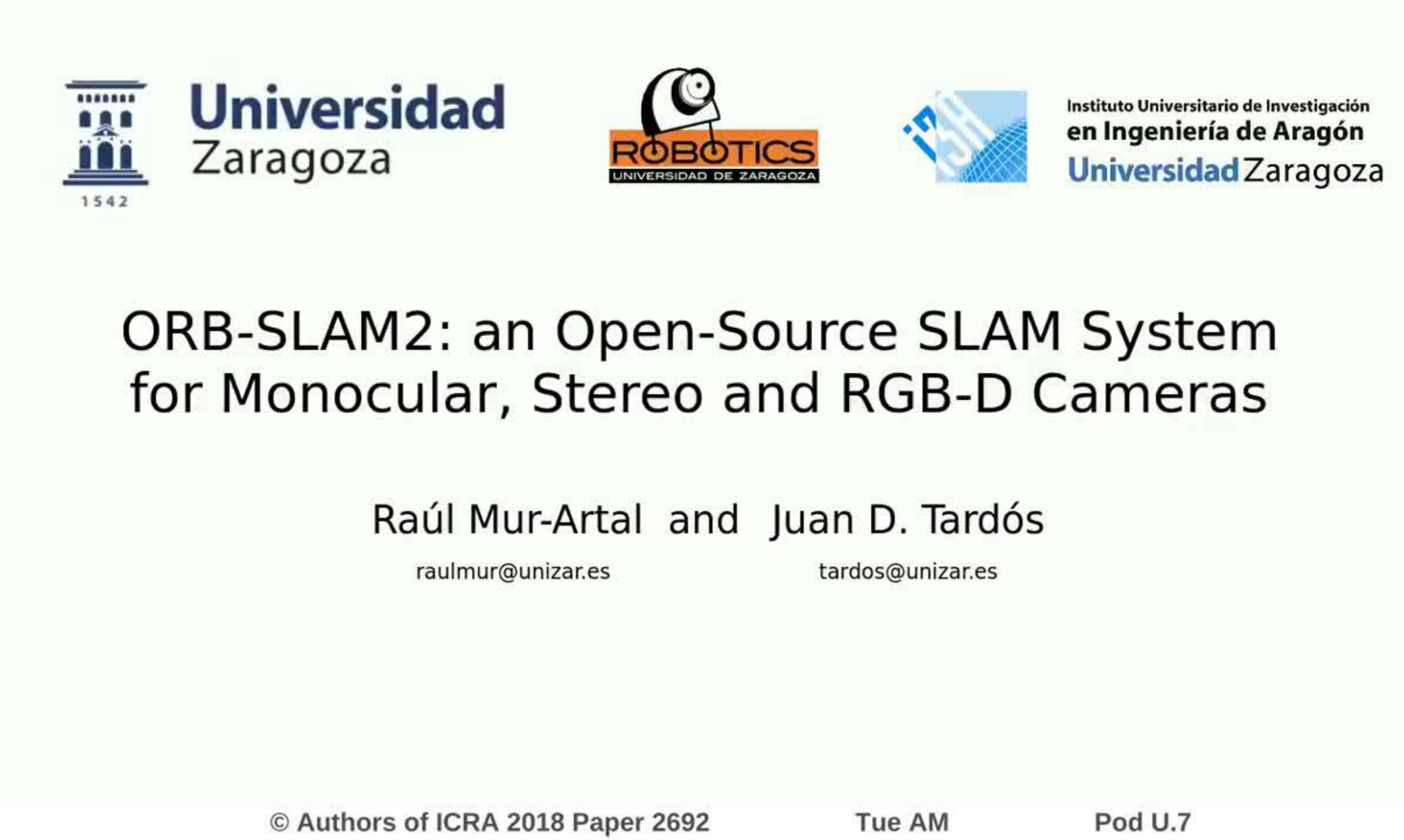
(Engel, Schöps, Cremers, ECCV 2014)

# Visual SLAM using Pose Graph Opt.

(Engel, Schöps, Cremers, ECCV 2014)

https://www.youtube.com/watch?time_continue=22&v=aBVXfqumTXc&feature=emb_logo

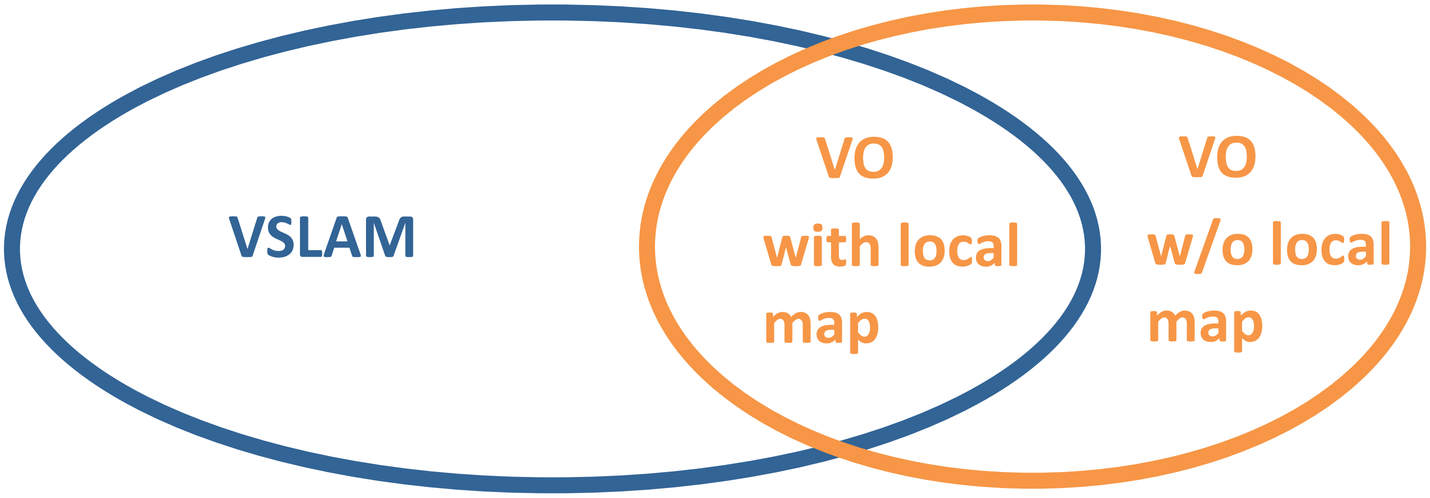
# Visual SLAM using Bundle Adjustment




(Mur-Artal, Tardos, T-RO 2017)

https://www.youtube.com/watch?v=ufvPS5wJAx0

# VO vs. VSLAM



# Structure from Motion

- Structure from Motion (SfM) denotes the joint estimation of
  - Structure, i.e. 3D reconstruction, and
  - Motion, i.e. 6-DoF camera poses,

  from a collection (i.e. unordered set) of images

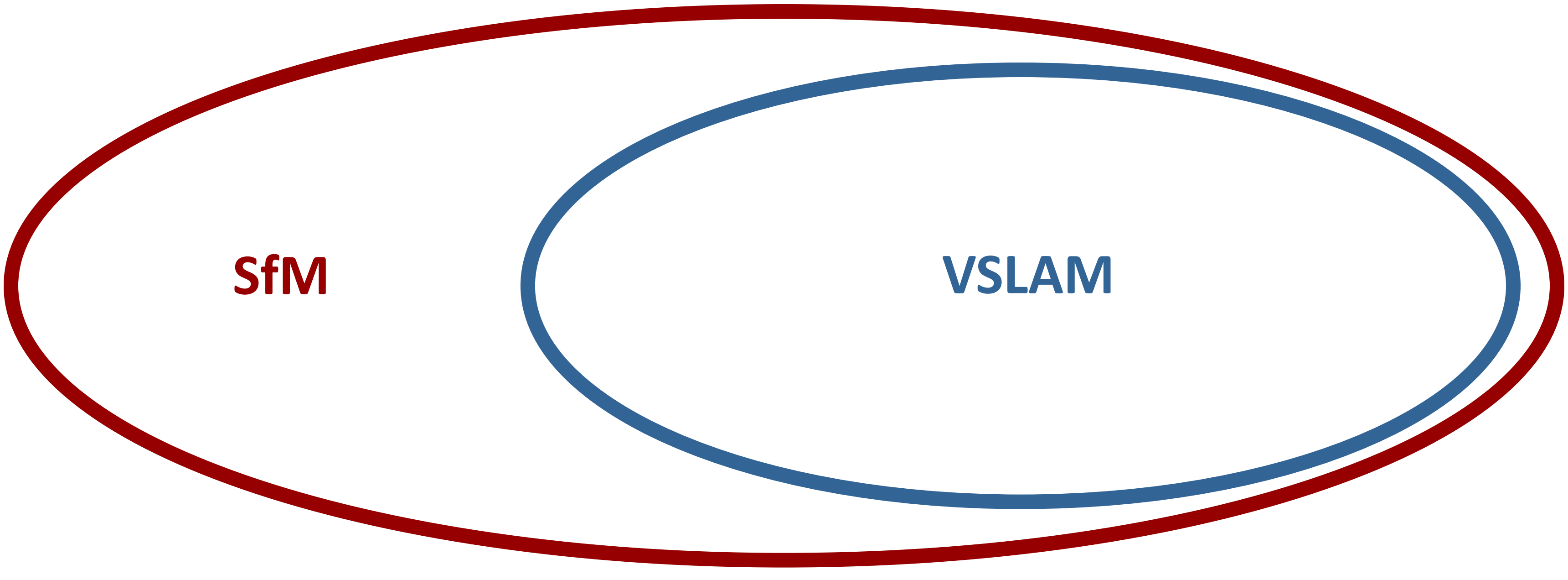
- Typical approach: keypoint matching and bundle adjustment

# Structure from Motion

(Schönberger, Frahm, CVPR 2016)

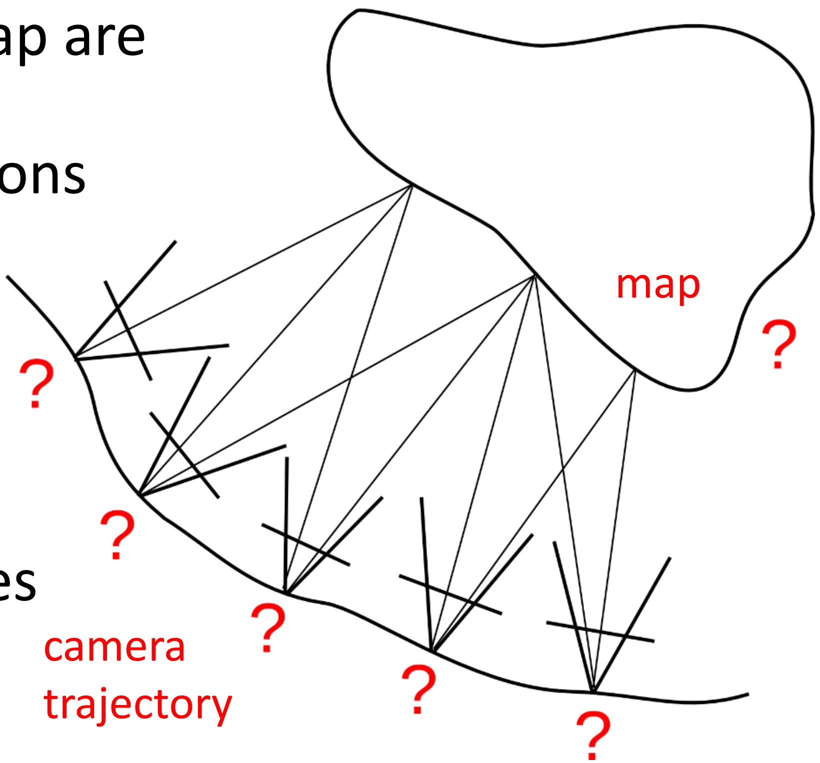
https://www.youtube.com/watch?v=PmXqdfBQxfQ

# VSLAM vs. SfM



# Why is SLAM difficult?

- Chicken-or-egg problem
  - Camera trajectory and map are unknown and need to be estimated from observations
  - Accurate localization requires an accurate map
  - Accurate mapping requires accurate localization




- How can we solve this problem efficiently and robustly?

# Definition of Visual SLAM

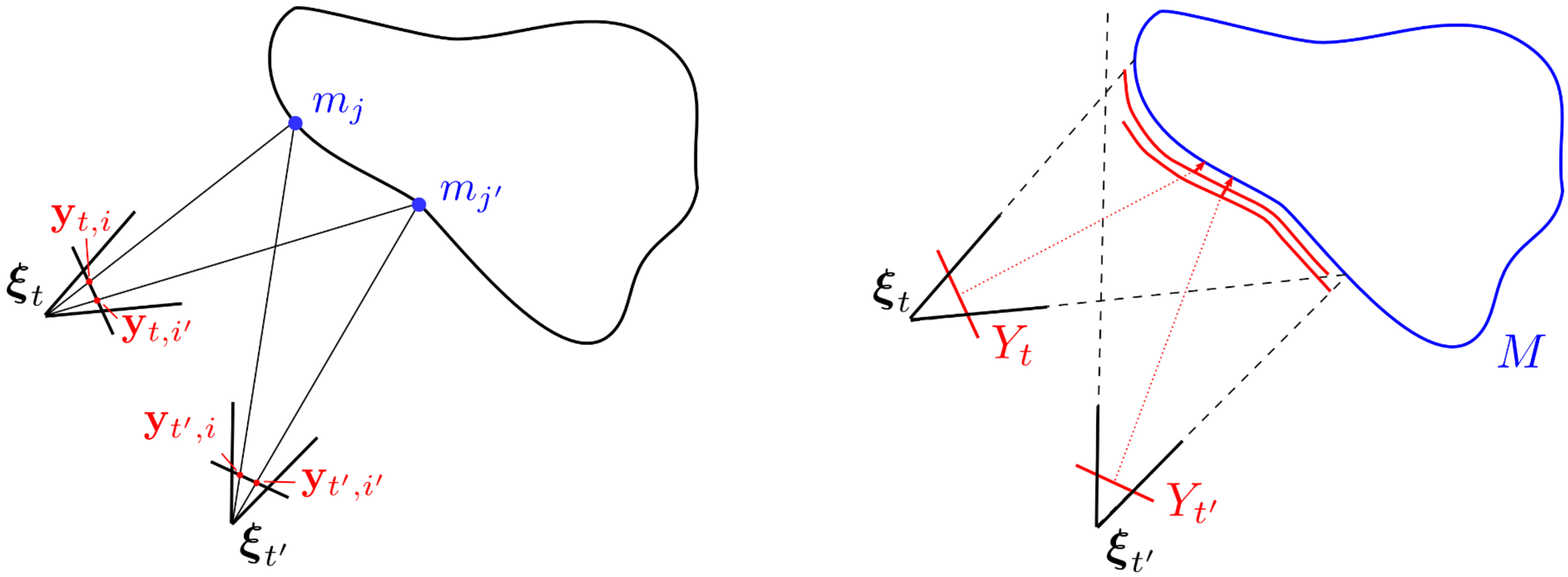
- Visual SLAM is the process of simultaneously estimating the egomotion of an object and the environment map using only inputs from visual sensors on the object and (if available) control inputs

- **Inputs:** images at discrete time steps $t$,
  - Monocular case: Set of images $I_{0:t} = \{I_0, \ldots, I_t\}$
  - Stereo case: Left/right images $I^l_{0:t} = \{I^l_0, \ldots, I^l_t\}$ $I^r_{0:t} = \{I^r_0, \ldots, I^r_t\}$
  - RGB-D case: Color/depth images $I_{0:t} = \{I_0, \ldots, I_t\}$ $Z_{0:t} = \{Z_0, \ldots, Z_t\}$

  - Robotics: **control inputs** $U_{1:t}$ often not considered/available

- **Output**:
  - Camera pose estimates $\mathbf{T}_t \in \mathbf{SE}(\mathbf{3})$ in world reference frame.
    For convenience, we also write $\boldsymbol{\xi}_t = \boldsymbol{\xi}\left(\mathbf{T}_t\right)$
    - reference frame is in general anchored at the first frame avalibale
  - Environment map $M$

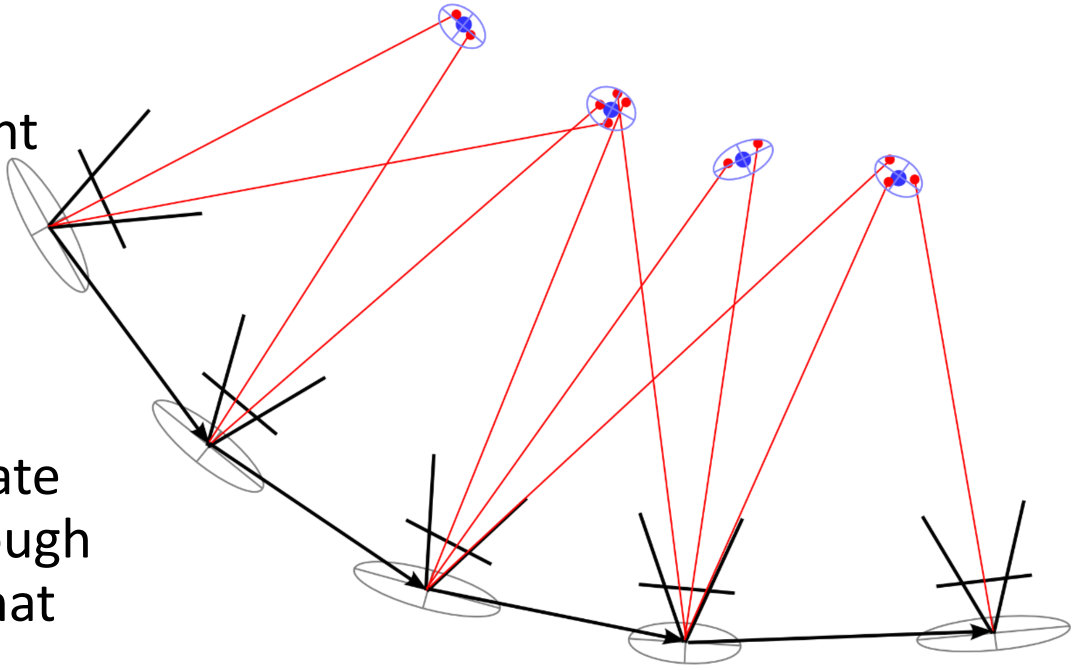
# Map Observations in Visual SLAM



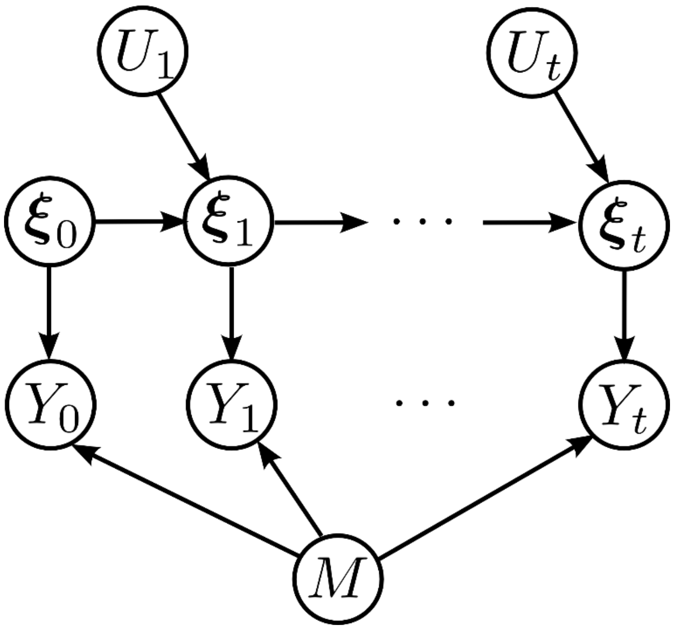

- With $Y_t$ we denote observations of the environment map in image $I_t$, f.e.
  - Indirect point-based method: $Y_t = \{\mathbf{y}_{t,1}, \ldots, \mathbf{y}_{t,N}\}$ (2D or 3D image points)
  - Direct method (mono): $Y_t = \{I_t\}$ (set of image pixels)
  - Direct RGB-D method: $Y_t = \{I_t, Z_t\}$ (all image pixels)
  - ...
- Involves data association to map elements $M = \{m_1, \ldots, m_S\}$
  - We denote correspondences by $c_{t,i} = j, 1 \leq i \leq N, 1 \leq j \leq S$

# SLAM Graph Optimization

- Joint optimization for poses and map elements from image observations of map elements
  - Common map element observations induce constraints between the poses
  - Map elements correlate with each others through the common poses that observe them
  - Bundle Adjustment



# Probabilistic Formulation of Visual SLAM




- SLAM posterior probability: $p\left(\boldsymbol{\xi}_{0:t}, M \mid Y_{0:t}, U_{1:t}\right)$
- Observation likelihood: $p\left(Y_t \mid \boldsymbol{\xi}_t, M\right)$
- State-transition probability: $p\left(\boldsymbol{\xi}_t \mid \boldsymbol{\xi}_{t-1}, U_t\right)$

# Probabilistic Formulation

- SLAM posterior: $p\left(\boldsymbol{\xi}_{0:t}, M \mid Y_{0:t}, U_{1:t}, c_{0:t}\right)$

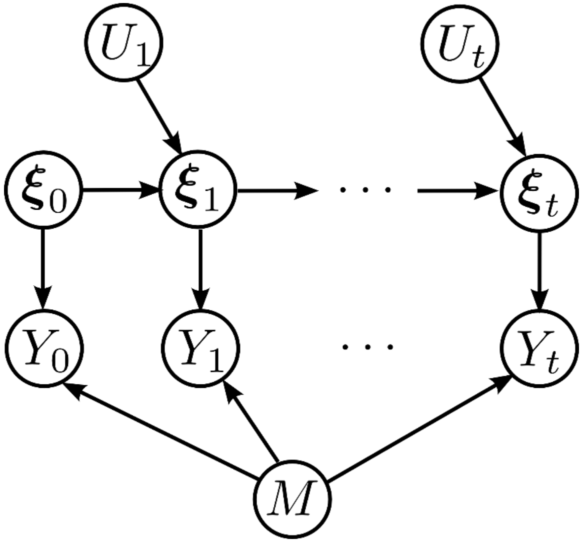
- Observation likelihood:

$$p\left(Y_t \mid \boldsymbol{\xi}_t, M, c_t\right) = p\left(Y_t \mid \boldsymbol{\xi}_t, m_{c_t}\right)$$
$$p(Y_t \mid \boldsymbol{\xi}_t, m_{c_t}) = \prod_i p(\mathbf{y}_{t,i} \mid \boldsymbol{\xi}_t, m_{c_{t,i}})$$

- State-transition probability:

$$p\left(\boldsymbol{\xi}_t \mid \boldsymbol{\xi}_{t-1}, U_t\right)$$

- SLAM posterior can be factorized



# Probabilistic Formulation

SLAM posterior factorization

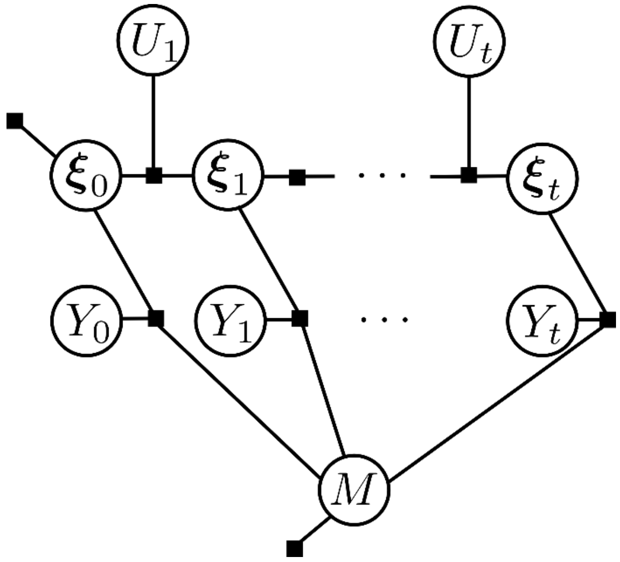
$$p(\boldsymbol{\xi}_{0:t}, M | Y_{0:t}, U_{1:t}, c_{0:t})$$
$$= \eta p(Y_t | \boldsymbol{\xi}_{0:t}, M, Y_{0:t-1}, U_{1:t}, c_{0:t}) p(\boldsymbol{\xi}_{0:t}, M | Y_{0:t-1}, U_{1:t}, c_{0:t})$$
$$= \eta p\left(Y_t \middle| \boldsymbol{\xi}_t, m_{c_t}\right) p(\boldsymbol{\xi}_{0:t}, M | Y_{0:t-1}, U_{1:t}, c_{0:t-1})$$
$$= \eta p\left(Y_t \middle| \boldsymbol{\xi}_t, m_{c_t}\right) p(\boldsymbol{\xi}_t | \boldsymbol{\xi}_{0:t-1}, M, Y_{0:t-1}, U_{1:t}, c_{0:t-1})$$
$$p(\boldsymbol{\xi}_{0:t-1}, M | Y_{0:t-1}, U_{1:t}, c_{0:t-1})$$
$$= \eta p\left(Y_t \middle| \boldsymbol{\xi}_t, m_{c_t}\right) p(\boldsymbol{\xi}_t | \boldsymbol{\xi}_{t-1}, U_t) p(\boldsymbol{\xi}_{0:t-1}, M | Y_{0:t-1}, U_{1:t}, c_{0:t-1})$$

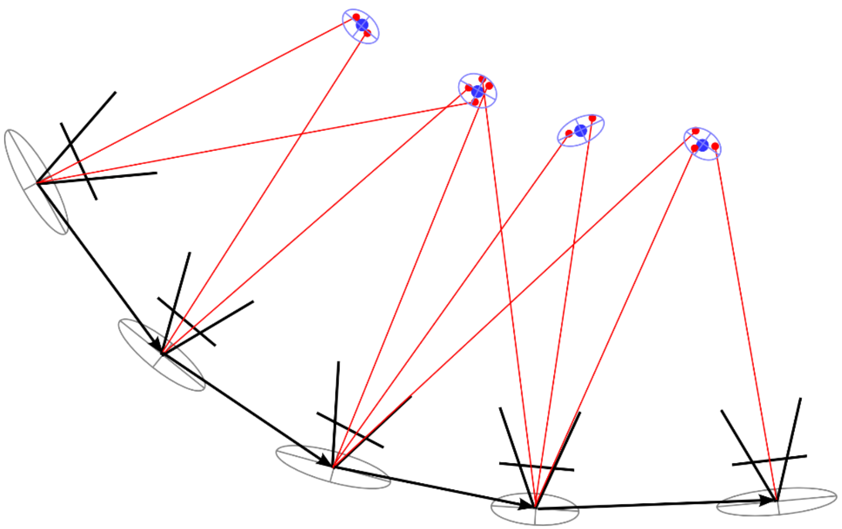
$$= \eta' p(\boldsymbol{\xi}_0) p(M) \prod_t p\left(Y_t \middle| \boldsymbol{\xi}_t, m_{c_t}\right) p(\boldsymbol{\xi}_t | \boldsymbol{\xi}_{t-1}, U_t)$$

# Factor Graph

- Factor graph representation of the full SLAM posterior

$$p\left(\boldsymbol{\xi}_{0:t}, M \mid Y_{0:t}, U_{1:t}, c_{0:t}\right)$$
$$= \eta \; p(\boldsymbol{\xi}_0)\, p(M) \prod_t p\left(Y_t \mid \boldsymbol{\xi}_t, m_{c_t}\right)\, p\left(\boldsymbol{\xi}_t \mid \boldsymbol{\xi}_{t-1}, U_t\right)$$



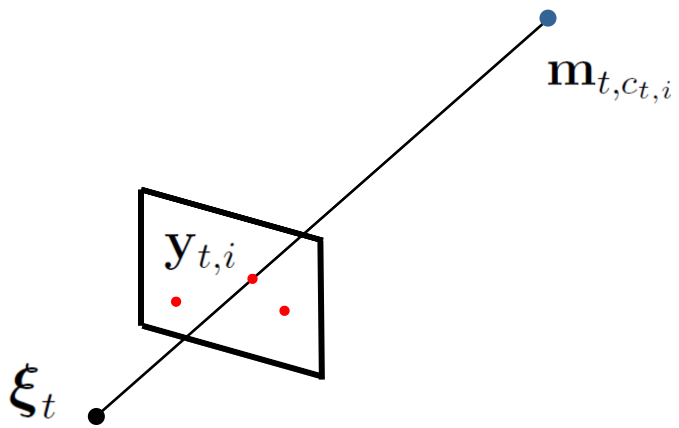



# Explicit Model

- $N_t$ noisy 2D point observation of 3D landmarks in each image, known data association

$$\mathbf{y}_{t,i} = h(\boldsymbol{\xi}_t, \mathbf{m}_{t,c_{t,i}}) + \boldsymbol{\delta}_t = \pi\left(\mathbf{T}(\boldsymbol{\xi}_t)^{-1}\overline{\mathbf{m}}_{t,c_{t,i}}\right) + \boldsymbol{\delta}_{t,i}$$

$$\boldsymbol{\delta}_{t,i} \sim \mathcal{N}\left(\mathbf{0}, \Sigma_{\mathbf{y}_{t,i}}\right)$$




- No control inputs

- Gaussian prior on pose $\xi_0 \sim \mathcal{N}\left(\boldsymbol{\xi}^0, \Sigma_{0,\xi}\right)$

- No prior on landmarks

# Full SLAM Optimization as Energy Minimization

- Optimize negative log posterior probability (MAP estimation)

$$E(\boldsymbol{\xi}_{0:t}, M) = \frac{1}{2} \left(\boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0\right)^\top \boldsymbol{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \left(\boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0\right)$$

$$+ \frac{1}{2} \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \left(\mathbf{y}_{\tau,i} - h(\boldsymbol{\xi}_\tau, \mathbf{m}_{c_{\tau,i}})\right)^\top \boldsymbol{\Sigma}_{\mathbf{y}_{\tau,i}}^{-1} \left(\mathbf{y}_{\tau,i} - h(\boldsymbol{\xi}_\tau, \mathbf{m}_{c_{\tau,i}})\right)$$

- Non-linear least squares!! We know how to optimize this.

- Remark: noisy state transitions based on control inputs add further residuals between subsequent poses

# Full SLAM Optimization as Energy Minimization

- Let‘s define the residuals on the full state vector

$$\mathbf{r}^0(\mathbf{x}) := \boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0 \qquad \text{(6D vector)}$$

$$\mathbf{r}^y_{t,i}(\mathbf{x}) := \mathbf{y}_{t,i} - h(\boldsymbol{\xi}_t, \mathbf{m}_{c_{t,i}}) \qquad \text{(2D vector)}$$

$$\mathbf{x} := \begin{pmatrix} \boldsymbol{\xi}_0 \\ \vdots \\ \boldsymbol{\xi}_t \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_S \end{pmatrix}$$

- Stack the residuals in a vector-valued function and collect the residual covariances on the diagonal blocks of a square matrix

$$\mathbf{r}(\mathbf{x}) := \begin{pmatrix} \mathbf{r}^0(\mathbf{x}) \\ \mathbf{r}^y_{0,1}(\mathbf{x}) \\ \vdots \\ \mathbf{r}^y_{t,N_t}(\mathbf{x}) \end{pmatrix} \qquad \mathbf{W} := \begin{pmatrix} \boldsymbol{\Sigma}^{-1}_{0,\boldsymbol{\xi}} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}^{-1}_{\mathbf{y}_{0,1}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{\Sigma}^{-1}_{\mathbf{y}_{t,N_t}} \end{pmatrix}$$

- Rewrite error function as $E(\mathbf{x}) = \frac{1}{2}\mathbf{r}(\mathbf{x})^\top \mathbf{W}\mathbf{r}(\mathbf{x})$

# Recap: Gauss-Newton Method

- Idea: Approximate Newton's method to minimize E(x)
  - Approximate E(x) through linearization of residuals

$$
\begin{aligned}
\widetilde{E}(\mathbf{x}) &= \frac{1}{2}\widetilde{\mathbf{r}}(\mathbf{x})^\top \mathbf{W}\widetilde{\mathbf{r}}(\mathbf{x}) \\
&= \frac{1}{2}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x}-\mathbf{x}_k\right)\right)^\top \mathbf{W}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x}-\mathbf{x}_k\right)\right) \qquad \mathbf{J}_k := \nabla_{\mathbf{x}}\mathbf{r}(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_k} \\
&= \frac{1}{2}\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W}\mathbf{r}(\mathbf{x}_k) + \underbrace{\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W}\mathbf{J}_k}_{=:\mathbf{b}_k^\top}\left(\mathbf{x}-\mathbf{x}_k\right) + \frac{1}{2}\left(\mathbf{x}-\mathbf{x}_k\right)^\top \underbrace{\mathbf{J}_k^\top \mathbf{W}\mathbf{J}_k}_{=:\mathbf{H}_k}\left(\mathbf{x}-\mathbf{x}_k\right)
\end{aligned}
$$

  - Find root of $\nabla_{\mathbf{x}}\widetilde{E}(\mathbf{x}) = \mathbf{b}_k^\top + \left(\mathbf{x}-\mathbf{x}_k\right)^\top \mathbf{H}_k$ using Newton's method, i.e.

$$
\nabla_{\mathbf{x}}\widetilde{E}(\mathbf{x}) = \mathbf{0} \text{ iff } \mathbf{x} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{b}_k
$$

- Pros:
  - Faster convergence (approx. quadratic convergence rate)
- Cons:
  - Divergence if too far from local optimum (H not positive definite)
  - Solution quality depends on initial guess

# Structure of the Bundle Adjustment Problem

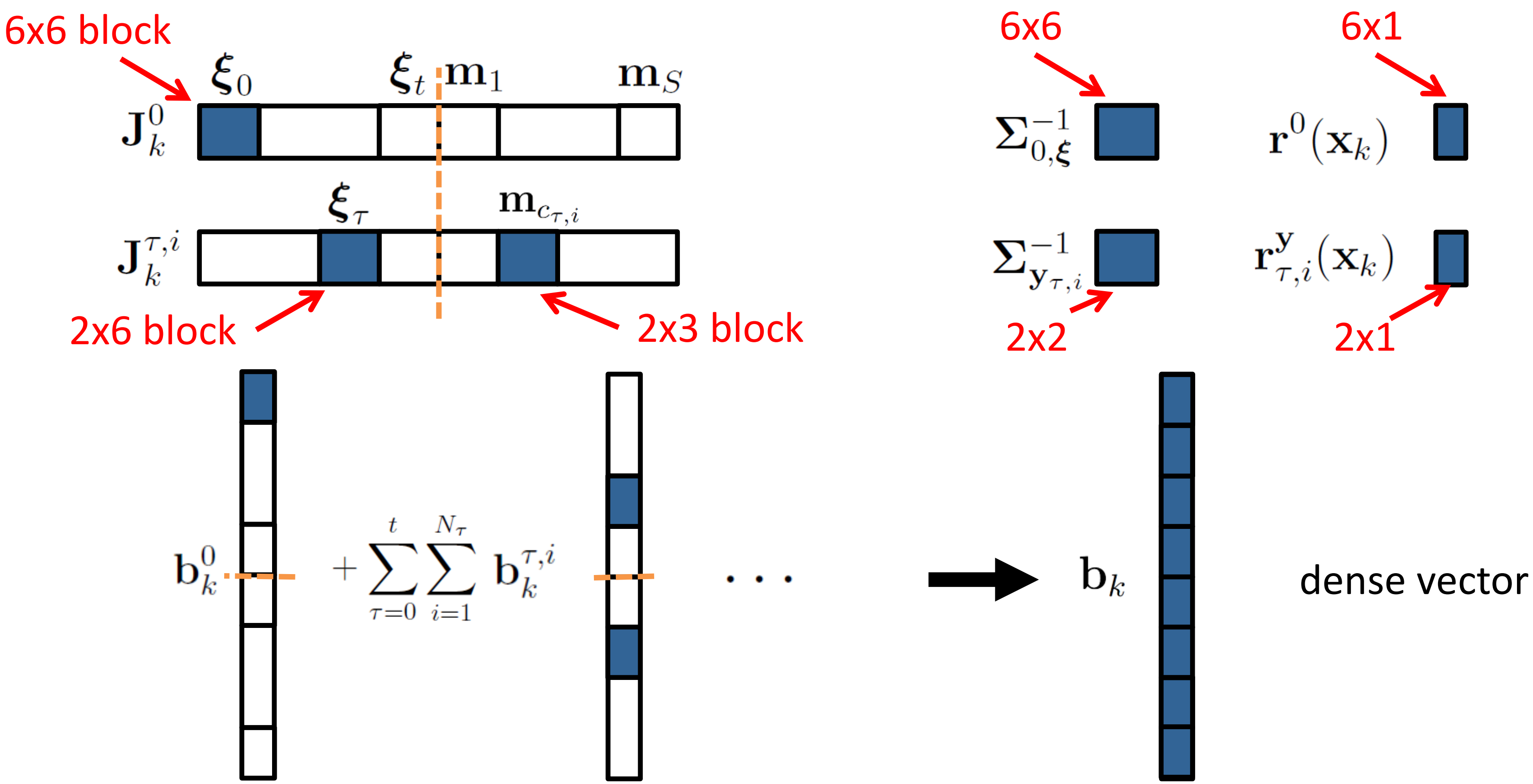
- $\mathbf{b}_k$ and $\mathbf{H}_k$ sum terms from individual residuals:

$$\mathbf{b}_k = \mathbf{b}_k^0 + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \mathbf{b}_k^{\tau,i} = \left(\mathbf{J}_k^0\right)^\top \mathbf{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \mathbf{r}^0(\mathbf{x}_k) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \left(\mathbf{J}_k^{\tau,i}\right)^\top \mathbf{\Sigma}_{\mathbf{y}_{\tau,i}}^{-1} \mathbf{r}_{\tau,i}^{\mathbf{y}}(\mathbf{x}_k)$$

$$\mathbf{H}_k = \mathbf{H}_k^0 + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \mathbf{H}_k^{\tau,i} = \left(\mathbf{J}_k^0\right)^\top \mathbf{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \left(\mathbf{J}_k^0\right) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \left(\mathbf{J}_k^{\tau,i}\right)^\top \mathbf{\Sigma}_{\mathbf{y}_{\tau,i}}^{-1} \left(\mathbf{J}_k^{\tau,i}\right)$$

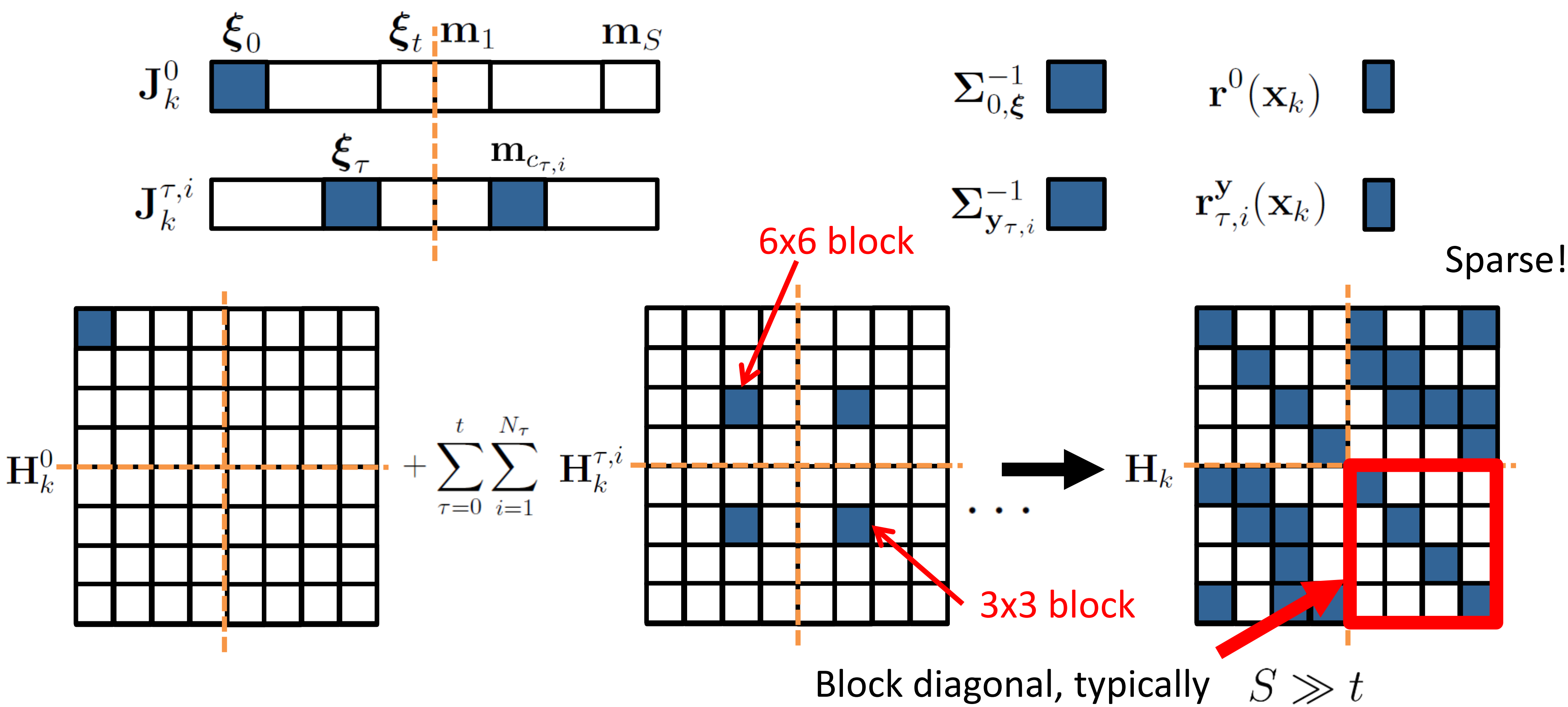
- What is the structure of these terms?

# Structure of the Bundle Adjustment Problem




$$\mathbf{b}_k = \mathbf{b}_k^0 + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \mathbf{b}_k^{\tau,i} = \left(\mathbf{J}_k^0\right)^\top \boldsymbol{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \mathbf{r}^0(\mathbf{x}_k) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \left(\mathbf{J}_k^{\tau,i}\right)^\top \boldsymbol{\Sigma}_{\mathbf{y}_{\tau,i}}^{-1} \mathbf{r}_{\tau,i}^{\mathbf{y}}(\mathbf{x}_k)$$

# Structure of the Bundle Adjustment Problem

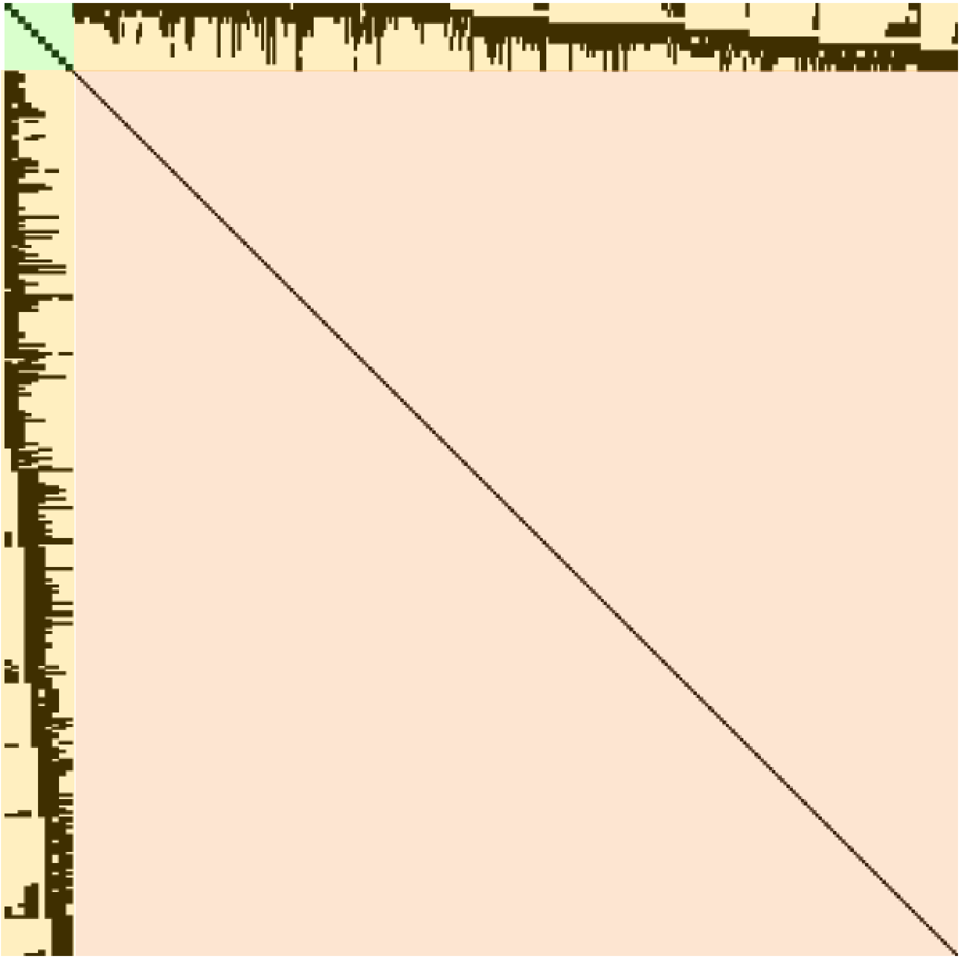



$$\mathbf{H}_k = \mathbf{H}_k^0 + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \mathbf{H}_k^{\tau,i} = \left(\mathbf{J}_k^0\right)^\top \boldsymbol{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \left(\mathbf{J}_k^0\right) + \sum_{\tau=0}^{t} \sum_{i=1}^{N_\tau} \left(\mathbf{J}_k^{\tau,i}\right)^\top \boldsymbol{\Sigma}_{\mathbf{y}_{\tau,i}}^{-1} \left(\mathbf{J}_k^{\tau,i}\right)$$

# Example Hessian of a BA Problem

Pose dimensions

(10 poses)



Landmark dimensions

(982 landmarks)

# Exploiting the Sparse Structure

- Idea:
  Apply the Schur complement to solve the system in a partitioned way

$$\mathbf{H}_k \Delta\mathbf{x} = -\mathbf{b}_k \quad \longrightarrow \quad \begin{pmatrix} \mathbf{H}_{\xi\xi} & \mathbf{H}_{\xi\mathrm{m}} \\ \mathbf{H}_{\mathrm{m}\xi} & \mathbf{H}_{\mathrm{mm}} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x}_\xi \\ \Delta\mathbf{x}_\mathrm{m} \end{pmatrix} = - \begin{pmatrix} \mathbf{b}_\xi \\ \mathbf{b}_\mathrm{m} \end{pmatrix}$$

$$\longrightarrow \quad \Delta\mathbf{x}_\xi = -\left(\mathbf{H}_{\xi\xi} - \mathbf{H}_{\xi\mathrm{m}}\mathbf{H}_{\mathrm{mm}}^{-1}\mathbf{H}_{\mathrm{m}\xi}\right)^{-1}\left(\mathbf{b}_\xi - \mathbf{H}_{\xi\mathrm{m}}\mathbf{H}_{\mathrm{mm}}^{-1}\mathbf{b}_\mathrm{m}\right)$$

$$\longrightarrow \quad \Delta\mathbf{x}_\mathrm{m} = -\mathbf{H}_{\mathrm{mm}}^{-1}\left(\mathbf{b}_\mathrm{m} + \mathbf{H}_{\mathrm{m}\xi}\Delta\mathbf{x}_\xi\right)$$

- Is this any better?

# Exploiting the Sparse Structure

- What is the structure of the two sub-problems ?

- Poses: $\Delta \mathbf{x}_\xi = -\underline{\left(\mathbf{H}_{\xi\xi} - \mathbf{H}_{\xi \mathrm{m}} \mathbf{H}_{\mathrm{mm}}^{-1} \mathbf{H}_{\mathrm{m}\xi}\right)}^{-1} \underline{\left(\mathbf{b}_\xi - \mathbf{H}_{\xi \mathrm{m}} \mathbf{H}_{\mathrm{mm}}^{-1} \mathbf{b}_{\mathrm{m}}\right)}$

$$\mathbf{H}_{\xi\xi} - \mathbf{H}_{\xi \mathrm{m}} \mathbf{H}_{\mathrm{mm}}^{-1} \mathbf{H}_{\mathrm{m}\xi} = \mathbf{H}_{\xi\xi} - \sum_{j=1}^{S} \mathbf{H}_{\xi \mathrm{m}_j} \mathbf{H}_{\mathrm{m}_j \mathrm{m}_j}^{-1} \mathbf{H}_{\mathrm{m}_j \xi} \qquad \mathbf{b}_\xi - \mathbf{H}_{\xi \mathrm{m}} \mathbf{H}_{\mathrm{mm}}^{-1} \mathbf{b}_{\mathrm{m}} = \mathbf{b}_\xi - \sum_{j=1}^{S} \mathbf{H}_{\xi \mathrm{m}_j} \mathbf{H}_{\mathrm{m}_j \mathrm{m}_j}^{-1} \mathbf{b}_{\mathrm{m_j}}$$

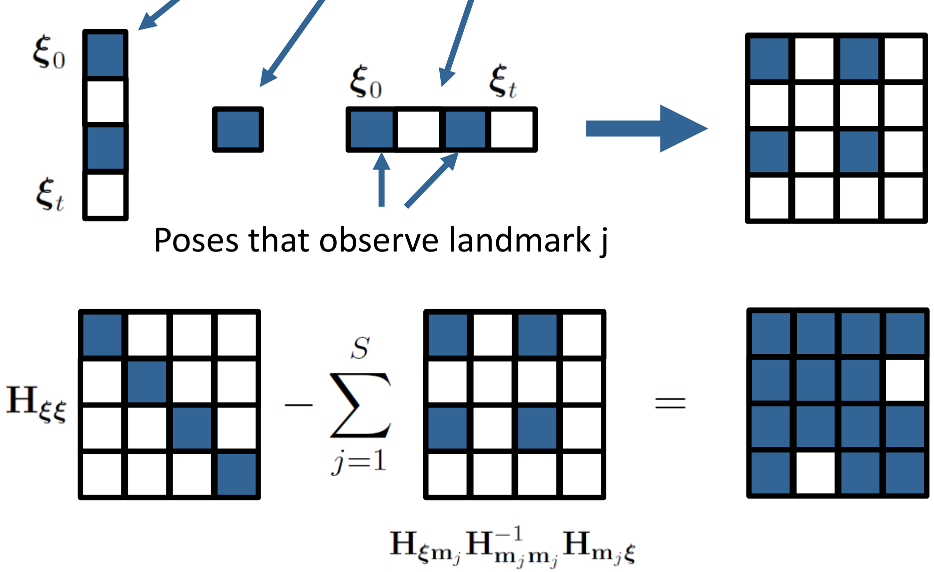
Reduced pose Hessian

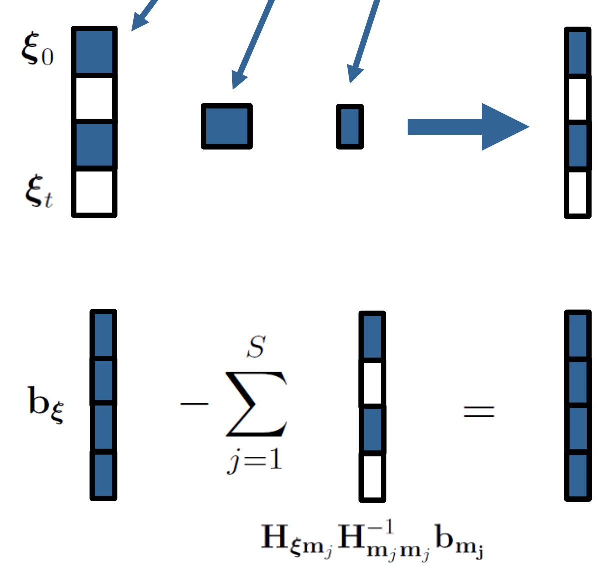
# Exploiting the Sparse Structure

- What is the structure of the two sub-problems ?
- Poses: $\Delta \mathbf{x}_\xi = -\left(\mathbf{H}_{\xi\xi} - \mathbf{H}_{\xi\mathbf{m}}\mathbf{H}_{\mathbf{mm}}^{-1}\mathbf{H}_{\mathbf{m}\xi}\right)^{-1}\left(\mathbf{b}_\xi - \mathbf{H}_{\xi\mathbf{m}}\mathbf{H}_{\mathbf{mm}}^{-1}\mathbf{b}_\mathbf{m}\right)$

$$\mathbf{H}_{\xi\xi} - \sum_{j=1}^{S} \mathbf{H}_{\xi\mathbf{m}_j}\mathbf{H}_{\mathbf{m}_j\mathbf{m}_j}^{-1}\mathbf{H}_{\mathbf{m}_j\xi}$$

$$\mathbf{b}_\xi - \sum_{j=1}^{S} \mathbf{H}_{\xi\mathbf{m}_j}\mathbf{H}_{\mathbf{m}_j\mathbf{m}_j}^{-1}\mathbf{b}_{\mathbf{m}_j}$$

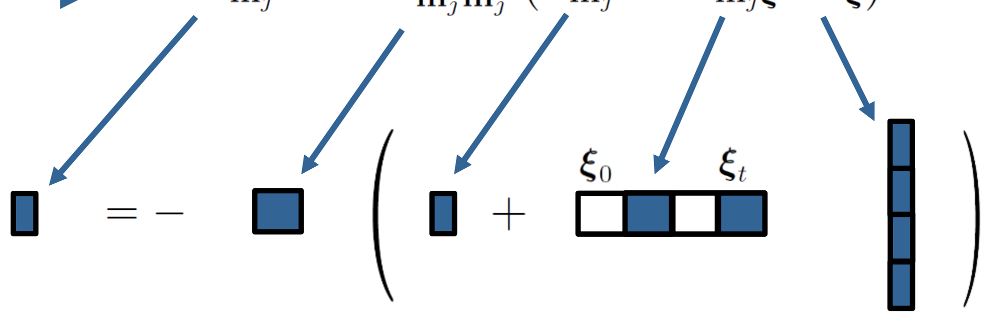


# Exploiting the Sparse Structure

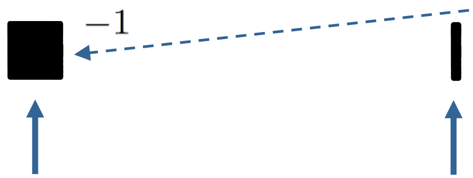
- What is the structure of the two sub-problems ?
- Landmarks: $\Delta \mathbf{x}_{\mathbf{m}} = -\mathbf{H}_{\mathbf{mm}}^{-1} \left( \mathbf{b}_{\mathbf{m}} + \mathbf{H}_{\mathbf{m}\xi} \Delta \mathbf{x}_{\xi} \right)$

$$\Delta \mathbf{x}_{\mathbf{m}_j} = -\mathbf{H}_{\mathbf{m}_j\mathbf{m}_j}^{-1} \left( \mathbf{b}_{\mathbf{m}_j} + \mathbf{H}_{\mathbf{m}_j\xi} \Delta \mathbf{x}_{\xi} \right)$$




  - Landmark-wise solution
  - Comparably small matrix operations
  - Only involves poses that observe the landmark

# Exploiting the Sparse Structure



$$\Delta \mathbf{x}_\xi = -\left(\mathbf{H}_{\xi\xi} - \mathbf{H}_{\xi \mathrm{m}} \mathbf{H}_{\mathrm{mm}}^{-1} \mathbf{H}_{\mathrm{m}\xi}\right)^{-1} \left(\mathbf{b}_\xi - \mathbf{H}_{\xi \mathrm{m}} \mathbf{H}_{\mathrm{mm}}^{-1} \mathbf{b}_{\mathrm{m}}\right)$$

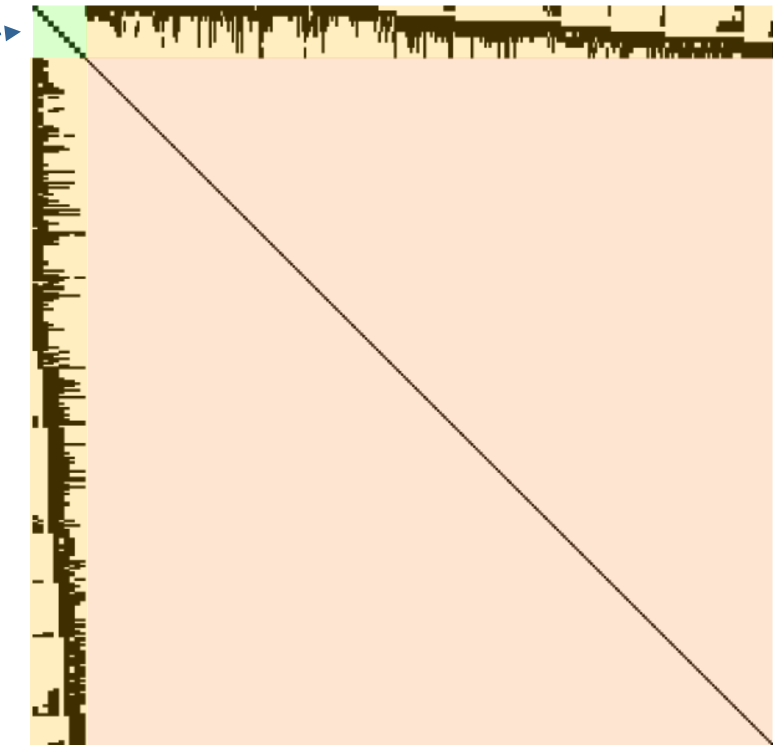
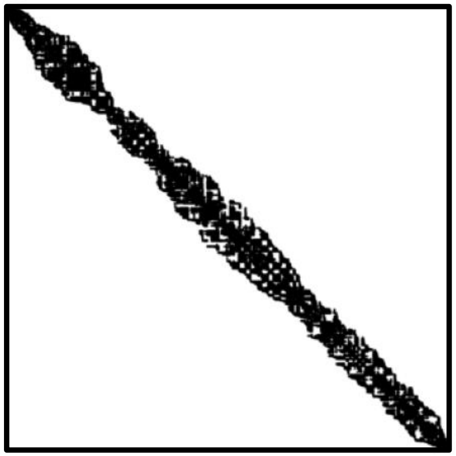
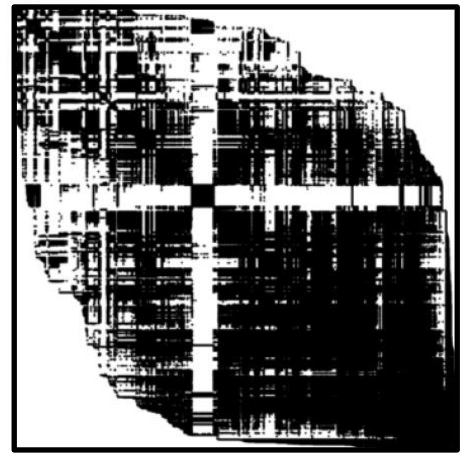


Image source: Manolis Lourakis (CC BY 3.0)

# Exploiting the Sparse Structure



Camera on a moving vehicle
(6375 images)

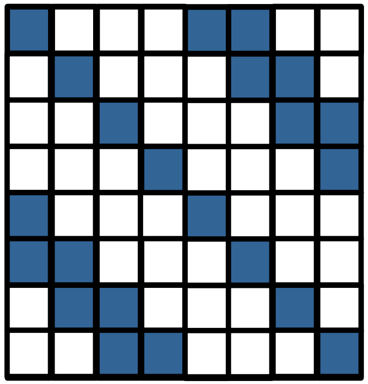


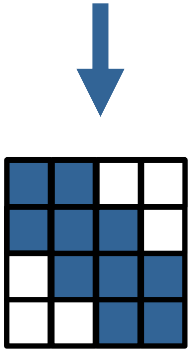
Flickr image search „Dubrovnik“
(4585 images)

- Reduced pose Hessian can still have sparse structure
- However: For many camera poses with many shared observations, the inversion of the reduced pose Hessian is still computationally expensive!

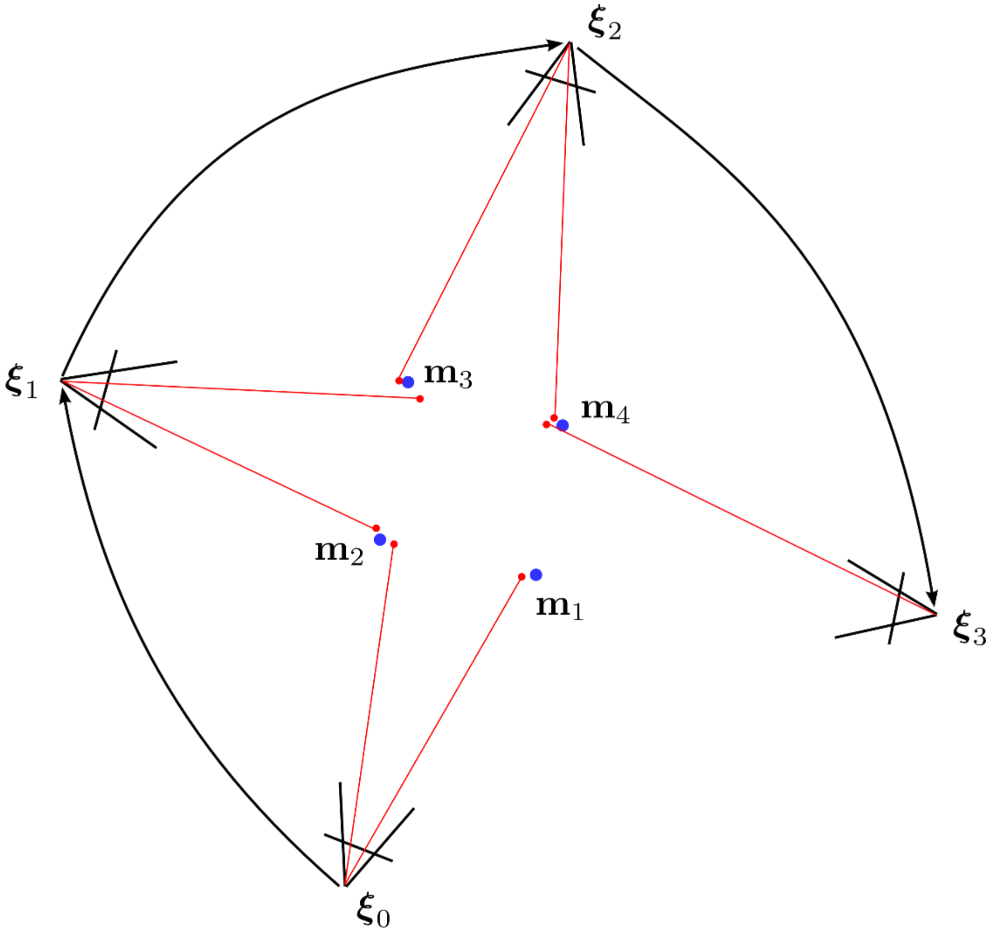
Image from Agarwal et al., ICCV 2009

# Effect of Loop-Closures on the Hessian

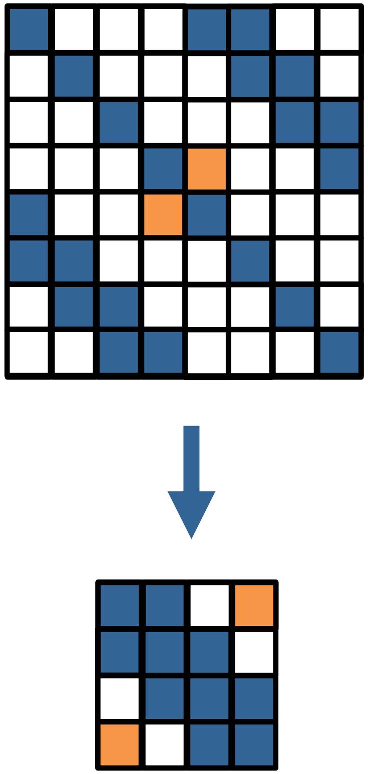




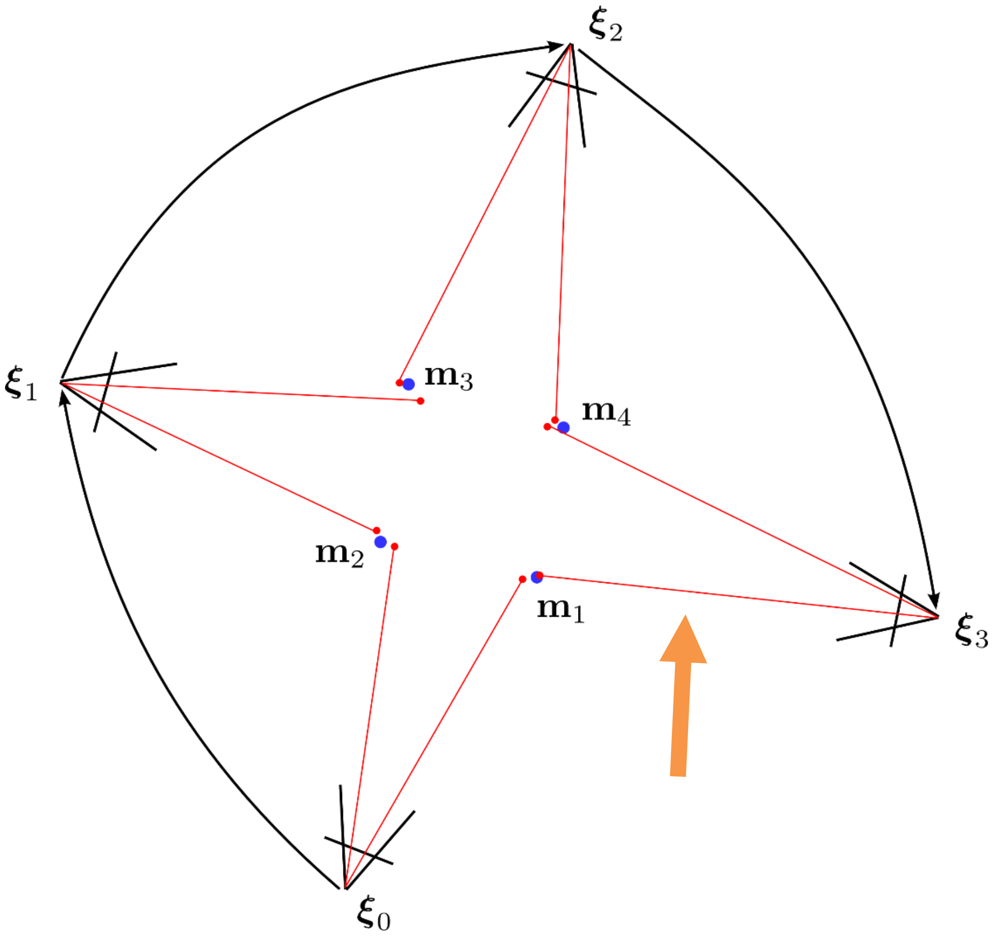
Band matrix



# Effect of Loop-Closures on the Hessian



Not band matrix: costlier to solve

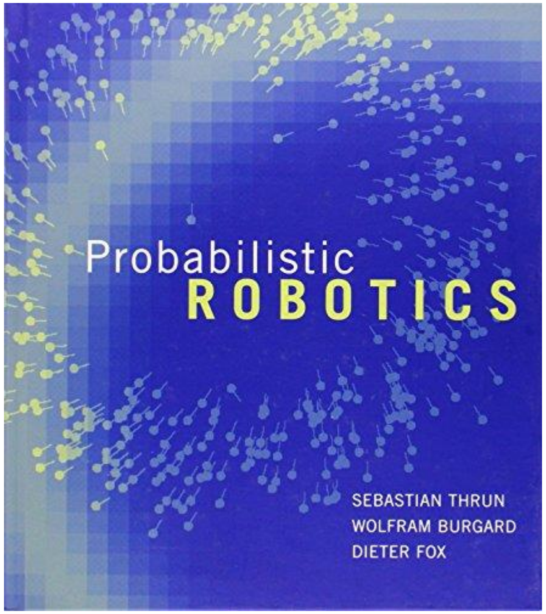


# Lessons Learned Today

- SLAM is a chicken-or-egg problem:
  - Localization requires map
  - Mapping requires localization
  - Unknown association of measurements to map elements
- Bundle Adjustment has a sparse structure that can be exploited for efficient optimization
- Reduction of BA to pose optimization problem through marginalization of landmarks (using the Schur complement)
- Loop closure constraints make SLAM optimization problem less efficient to solve (but reduce drift!)
  - In general does not have to be solved in real-time

# Further Reading

- Probabilistic Robotics textbook




Probabilistic Robotics,
S. Thrun, W. Burgard, D. Fox,
MIT Press, 2005

- Triggs et al., Bundle Adjustment – A Modern Synthesis, 2002

Thanks for your attention!

# Slides Information

- These slides have been initially created by Jörg Stückler as part of the lecture "Robotic 3D Vision" in winter term 2017/18 at Technical University of Munich.
- The slides have been revised by myself (Niclas Zeller) for the same lecture held in winter term 2020/21
- Acknowledgement of all people that contributed images or video material has been tried (please kindly inform me if such an acknowledgement is missing so it can be added).