Robotic 3D Vision

Lecture 12: Visual SLAM 3 – Pose Graph Optimization, Place Recognition

WS 2020/21

Dr. Niclas Zeller

Artisense GmbH

What We Will Cover Today

- Hybrid SLAM methods
- Pose graph optimization
- Loop closure detection and place recognition

Recap: What is Visual SLAM ?

- SLAM stands for Simultaneous Localization and Mapping
  - Estimate the pose of the camera in a map, and simultaneously
  - Reconstruct the environment map
- Visual SLAM (VSLAM): SLAM with vision sensors
- Loop-closure: Revisiting a place allows for drift compensation

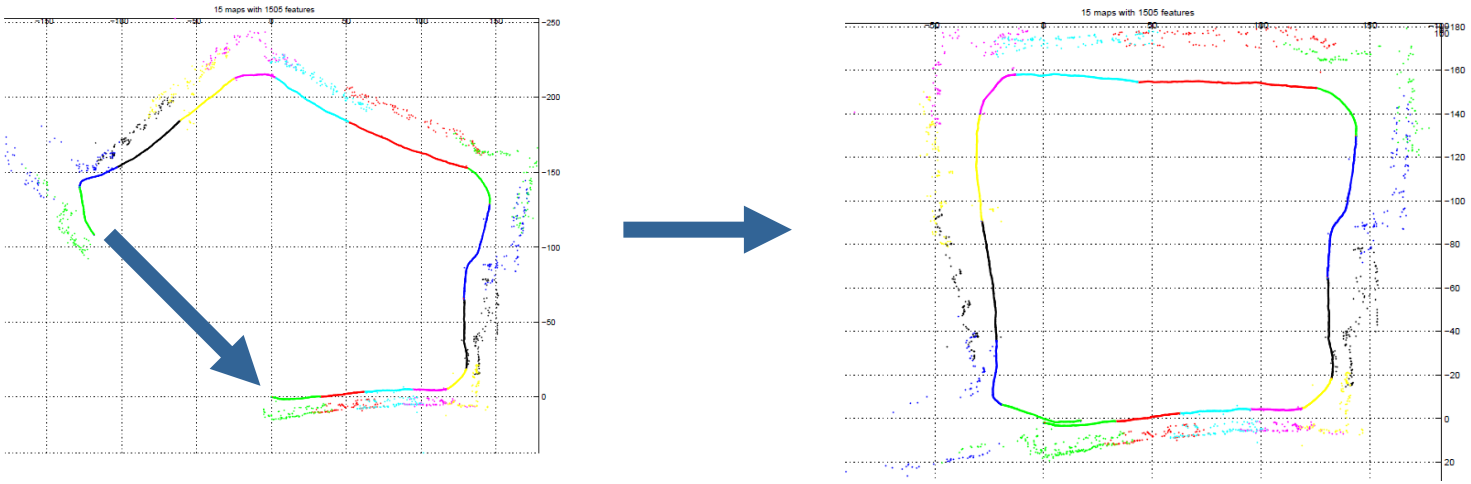
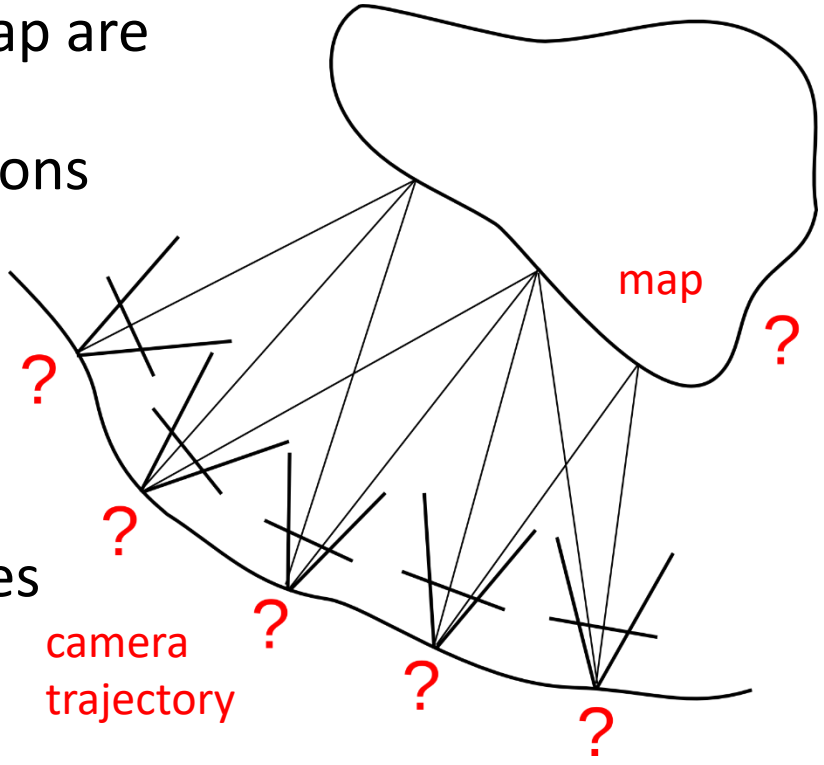


Image from Clemente et al., RSS 2007

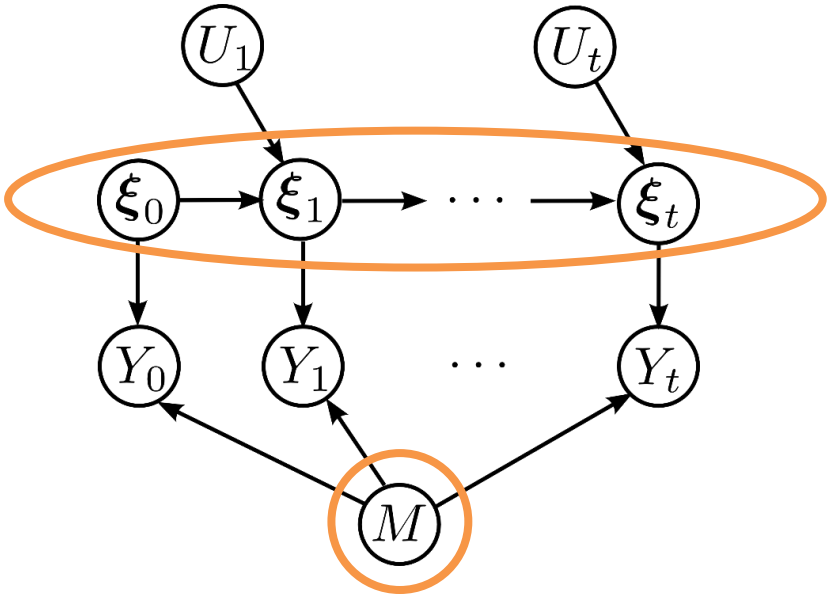
Recap: Why is SLAM difficult?

- 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?

Recap: Probabilistic Formulation of Visual SLAM



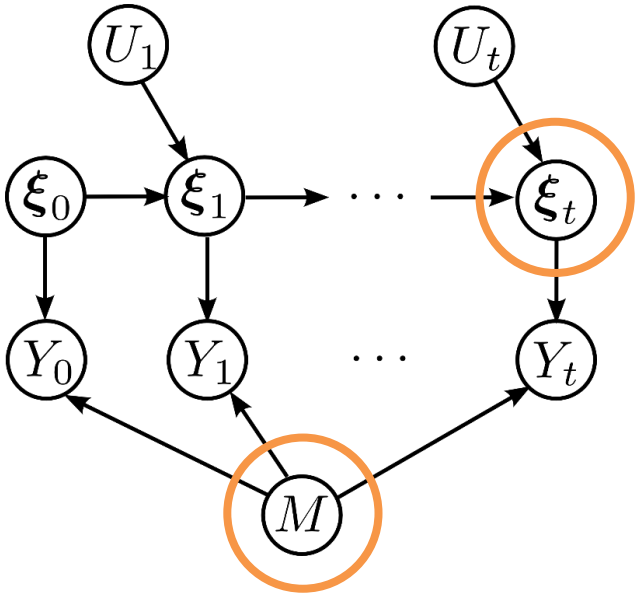
- SLAM posterior probability: $p\left(\boldsymbol{\xi}_{0:t}, M \mid Y_{0:t}, U_{1:t}\right)$
- Observation likelihood: $p\left(Y_t \mid \boldsymbol{\xi}_t, M\right)$
- State-transition probability: $p\left(\boldsymbol{\xi}_t \mid \boldsymbol{\xi}_{t-1}, U_t\right)$

Recap: Online SLAM Methods

- Marginalize out previous poses

$$p\left(\boldsymbol{\xi}_t, M \mid Y_{0:t}, U_{1:t}\right) =$$

$$\int \dots \int p\left(\boldsymbol{\xi}_{0:t}, M \mid Y_{0:t}, U_{1:t}\right) d\boldsymbol{\xi}_{t-1} \dots d\boldsymbol{\xi}_0$$



- Poses can be marginalized individually in a recursive way

- Variants:
  - Tracking-and-Mapping: Alternating pose and map estimation
  - Probabilistic filters, f.e. EKF-SLAM

Hybrid SLAM Methods

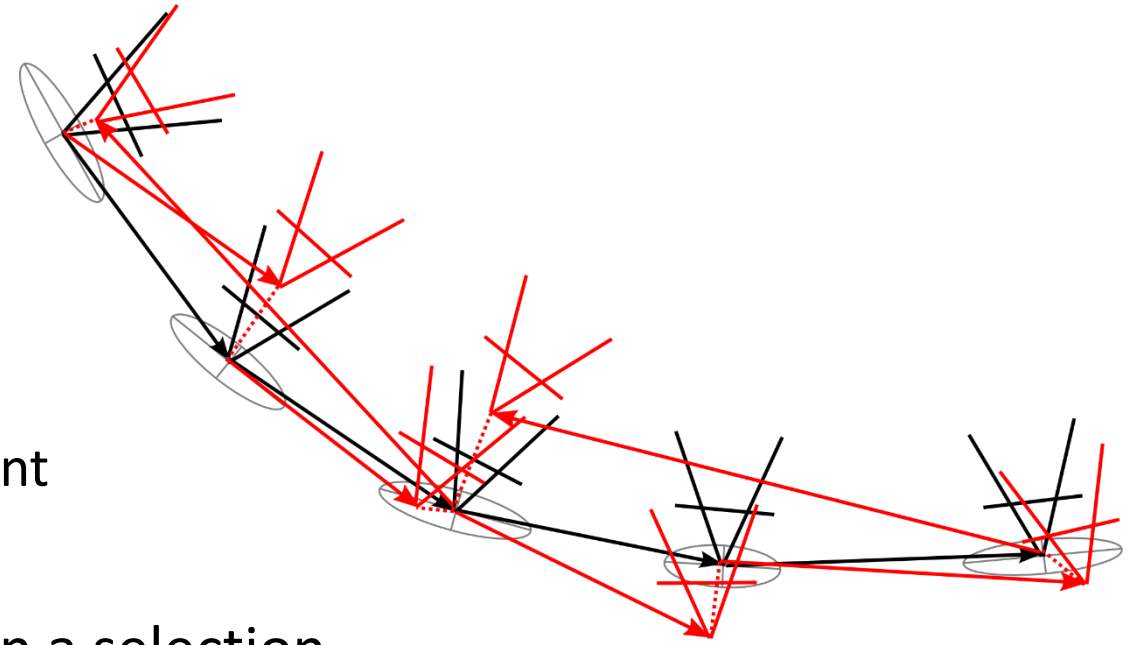
- Global vs. local optimization methods
  - Global
    - Full SLAM opt. (Bundle Adjustment (BA)), pose graph optimization (PGO), etc.
    - In general not real-time capable
  - Local
    - incremental tracking and local mapping
      - VO with local map (e.g. PnP + local BA, alternating tracking and mapping)
      - Designed for real-time operation
- Hybrid SLAM
  - Real-time local SLAM
  - Global optimization in a slower parallel process
    - LSD-SLAM, ORB-SLAM, etc.

- State-of-the-art SLAM methods are often running three parallel optimization threads
  - Real-time PnP/image alignment, local BA, global BA/PGO
  - ORB-SLAM, LDSO, etc.

Pose Graph Optimization (PGO)

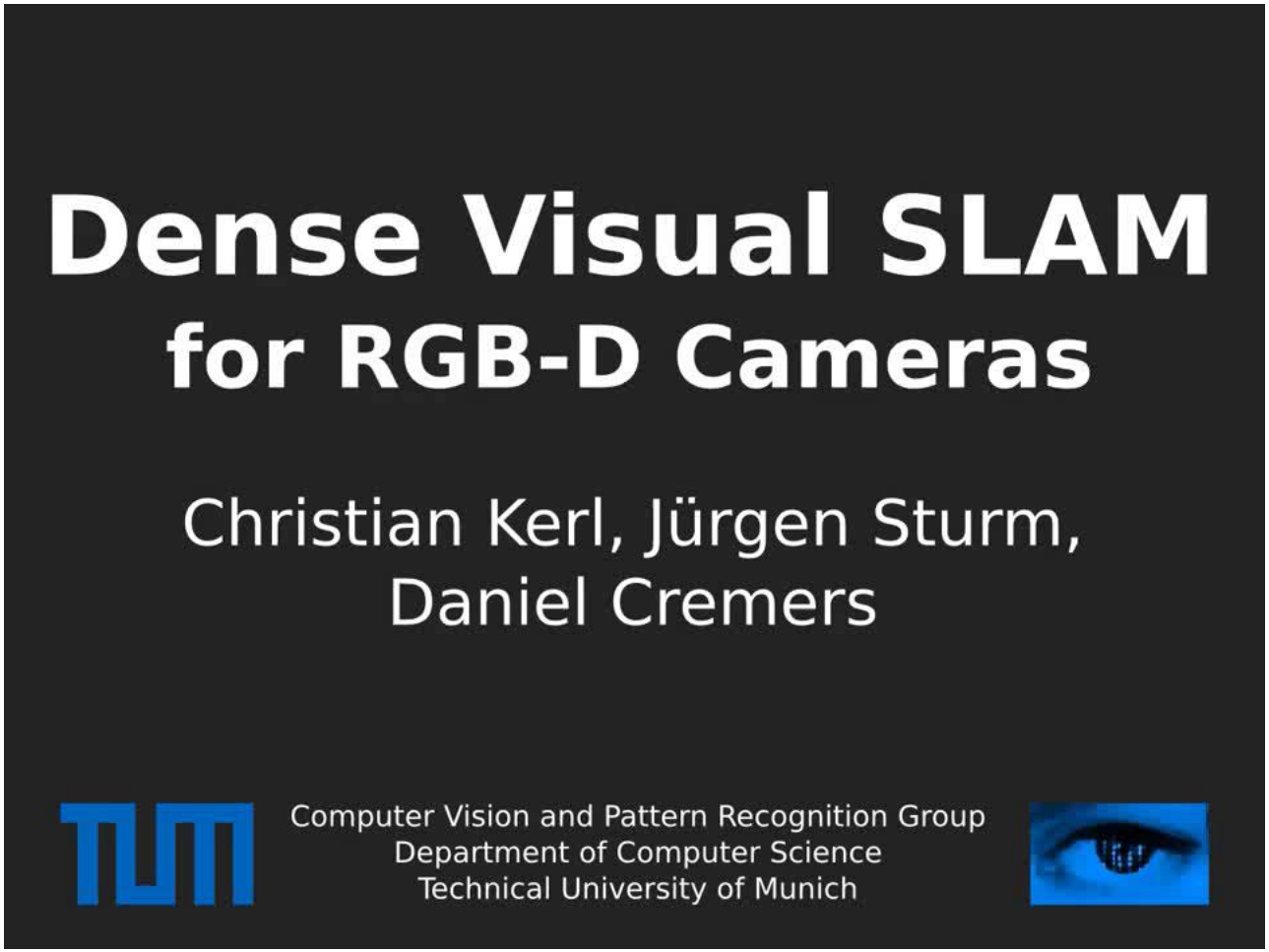
- Optimization of poses from relative pose constraints, map recovered from the optimized poses

- Obtain relative constraints between poses from image observations, e.g.
  - 8-point algorithm
  - Direct image alignment



- In gereral performed on a selection of keyframes

PGO Example (RGB-D SLAM)



(Kerl, Sturm, Cremers, IROS 2013)

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

Probabilistic Formulation of PGO

- Full SLAM reduced to trajectory optimization
  - Corresponds to marginalization of the map
  - Alternating optimization of reduced pose graph problem and map
    - Globally consistent map can also be obtained from local map segments and globally consistent poses

- Approximation to SLAM posterior distribution

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

optimize poses directly: $p\left(\boldsymbol{\xi}_{0:t} \mid \left\{\widetilde{\boldsymbol{\xi}}_i^j\right\}, U_{1:t}\right)$

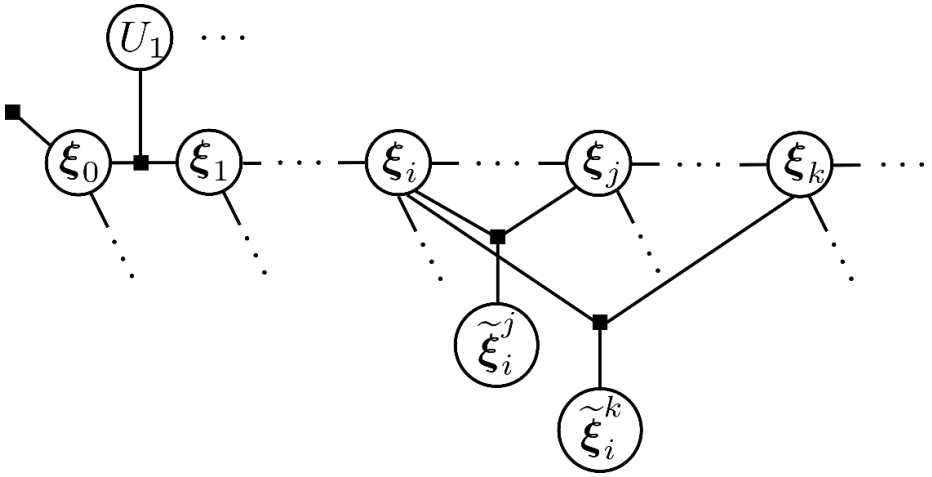
using probabilistic observations of relative poses that are estimated from the image observations Y_i, Y_j : $p\left(\widetilde{\boldsymbol{\xi}}_i^j \mid \boldsymbol{\xi}_i, \boldsymbol{\xi}_j\right)$

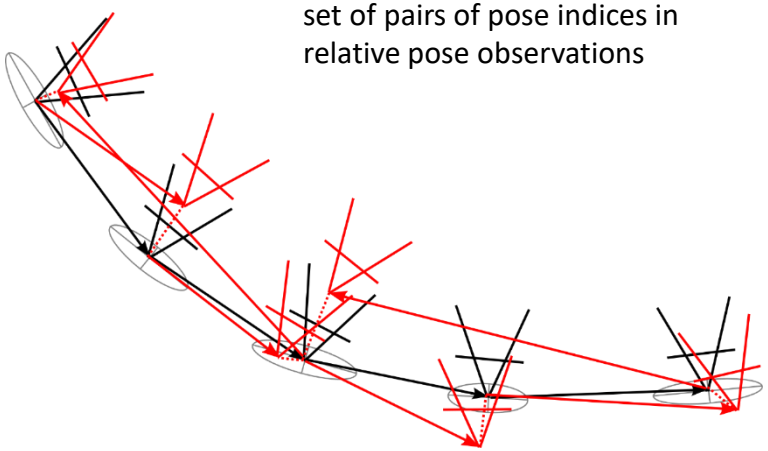
Factor Graph of PGO

- Factor graph representation of the relative pose graph formulation

$$p\left(\boldsymbol{\xi}_{0:t} \mid \left\{\widetilde{\boldsymbol{\xi}}_i^j, U_{1:t}\right\}\right) = \eta p\left(\boldsymbol{\xi}_0\right) \prod_\tau p(\boldsymbol{\xi}_\tau \mid \boldsymbol{\xi}_{\tau-1}, U_\tau) \prod_{(i,j)\in\mathcal{C}} p\left(\widetilde{\boldsymbol{\xi}}_i^j \mid \boldsymbol{\xi}_i, \boldsymbol{\xi}_j\right)$$

set of pairs of pose indices in relative pose observations





Recap: Some Notation for Twist Coordinates

- Let's define the following notation:
  - Inv. of hat operator: $$\begin{pmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}^\vee = (\omega_1 \ \omega_2 \ \omega_3 \ v_1 \ v_2 \ v_3)^\top$$
  - Conversion: $\boldsymbol{\xi}(\mathbf{T}) = (\log(\mathbf{T}))^\vee \qquad \mathbf{T}(\boldsymbol{\xi}) = \exp(\widehat{\boldsymbol{\xi}})$
  - Pose inversion: $\boldsymbol{\xi}^{-1} = \log(\mathbf{T}(\boldsymbol{\xi})^{-1})^\vee = -\boldsymbol{\xi}$
  - Pose concatenation: $\boldsymbol{\xi}_1 \oplus \boldsymbol{\xi}_2 = (\log(\mathbf{T}(\boldsymbol{\xi}_2)\,\mathbf{T}(\boldsymbol{\xi}_1)))^\vee$
  - Pose difference: $\boldsymbol{\xi}_1 \ominus \boldsymbol{\xi}_2 = \left(\log\left(\mathbf{T}(\boldsymbol{\xi}_2)^{-1}\,\mathbf{T}(\boldsymbol{\xi}_1)\right)\right)^\vee$

An Explicit Model for PGO

Pose difference:

$$\boldsymbol{\xi}_1 \ominus \boldsymbol{\xi}_2 = \left(\log\left(\mathbf{T}\left(\boldsymbol{\xi}_2\right)^{-1}\mathbf{T}\left(\boldsymbol{\xi}_1\right)\right)\right)^{\vee}$$

- Noisy observation of relative motion between camera poses

$$p\left(\widetilde{\boldsymbol{\xi}}_i^j \mid \boldsymbol{\xi}_i, \boldsymbol{\xi}_j\right) = \mathcal{N}\left(\left(\boldsymbol{\xi}_i \ominus \boldsymbol{\xi}_j\right) \ominus \widetilde{\boldsymbol{\xi}}_i^j; \mathbf{0}, \boldsymbol{\Sigma}_{i,j}\right)$$

- No control inputs available / no state-transition model

- Gaussian prior on pose $\boldsymbol{\xi}_0 \sim \mathcal{N}\left(\boldsymbol{\xi}^0, \boldsymbol{\Sigma}_{0,\xi}\right)$

Gaussian distribution on Lie Manifold

$$p(\boldsymbol{\xi}_0) = \mathcal{N}\left(\boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0; \mathbf{0}, \Sigma_{0,\xi}\right)$$
$$= \left((2\pi)^p \det \Sigma_{0,\xi}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0)^{\mathrm{T}}\Sigma_{0,\xi}^{-1}(\boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0)\right) \qquad p = \dim(\boldsymbol{\xi}_0)$$

PGO as Energy Minimization

- Optimize negative log posterior probability (MAP estimation)

$$E\left(\boldsymbol{\xi}_{0:t}\right) = \frac{1}{2}\left(\boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0\right)^\top \boldsymbol{\Sigma}_{0,\xi}^{-1}\left(\boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0\right) + \frac{1}{2}\sum_{(i,j)\in\mathcal{C}}\left(\left(\boldsymbol{\xi}_i \ominus \boldsymbol{\xi}_j\right) \ominus \widetilde{\boldsymbol{\xi}}_i^j\right)^\top \boldsymbol{\Sigma}_{i,j}^{-1}\left(\left(\boldsymbol{\xi}_i \ominus \boldsymbol{\xi}_j\right) \ominus \widetilde{\boldsymbol{\xi}}_i^j\right)$$

- Non-linear least squares…

PGO as Energy Minimization

- Let‘s define the residuals on the full state vector $\mathbf{x} := \begin{pmatrix} \boldsymbol{\xi}_0 \\ \vdots \\ \boldsymbol{\xi}_t \end{pmatrix}$

$$\mathbf{r}^0(\mathbf{x}) := \boldsymbol{\xi}_0 \ominus \boldsymbol{\xi}^0$$

$$\mathbf{r}^{i,j}(\mathbf{x}) := (\boldsymbol{\xi}_i \ominus \boldsymbol{\xi}_j) \ominus \widetilde{\boldsymbol{\xi}}_i^j$$

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

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

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

Structure of the PGO Problem

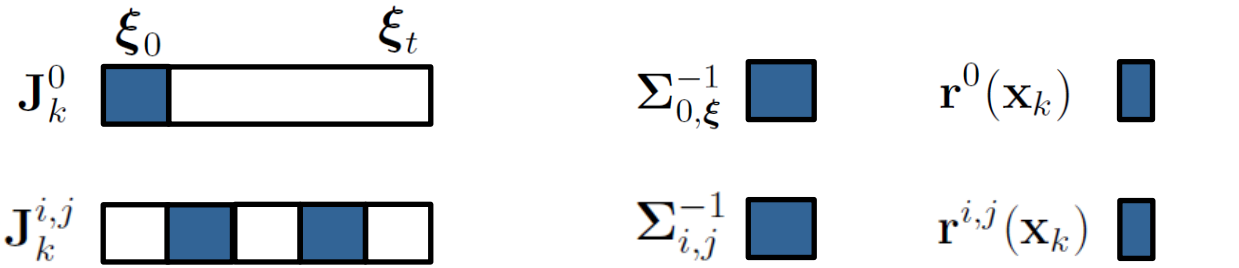
- Leads to $\mathbf{H}_k \Delta \mathbf{x} = -\mathbf{b}_k$ with

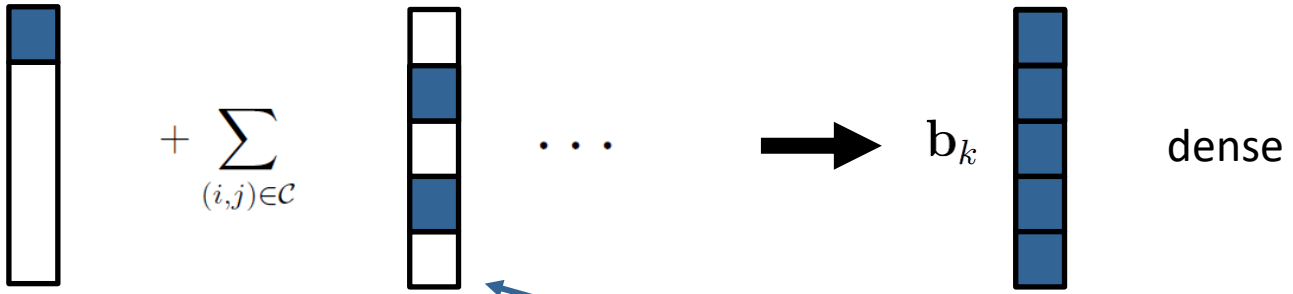
$$\mathbf{b}_k = \mathbf{J}_k^\top \mathbf{W} \mathbf{r}(\mathbf{x}) = \left(\mathbf{J}_k^0\right)^\top \boldsymbol{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \mathbf{r}^0(\mathbf{x}_k) + \sum_{(i,j)\in\mathcal{C}} \left(\mathbf{J}_k^{i,j}\right)^\top \boldsymbol{\Sigma}_{i,j}^{-1} \mathbf{r}^{i,j}(\mathbf{x}_k)$$

$$\mathbf{H}_k = \mathbf{J}_k^\top \mathbf{W} \mathbf{J}_k = \left(\mathbf{J}_k^0\right)^\top \boldsymbol{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \mathbf{J}_k^0 + \sum_{(i,j)\in\mathcal{C}} \left(\mathbf{J}_k^{i,j}\right)^\top \boldsymbol{\Sigma}_{i,j}^{-1} \mathbf{J}_k^{i,j}$$

- What is the structure now?

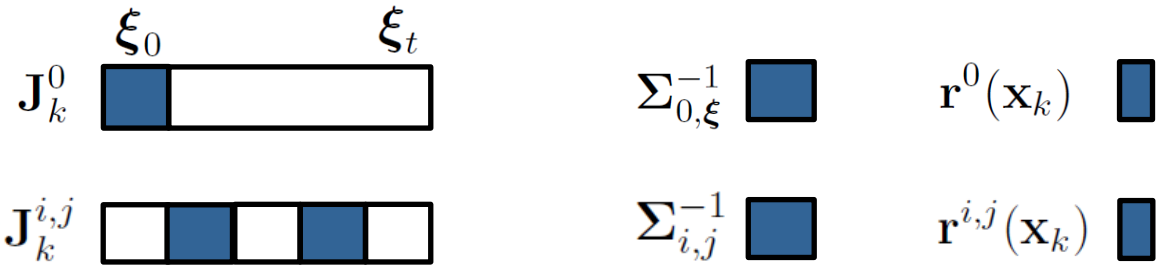
Structure of the PGO Problem

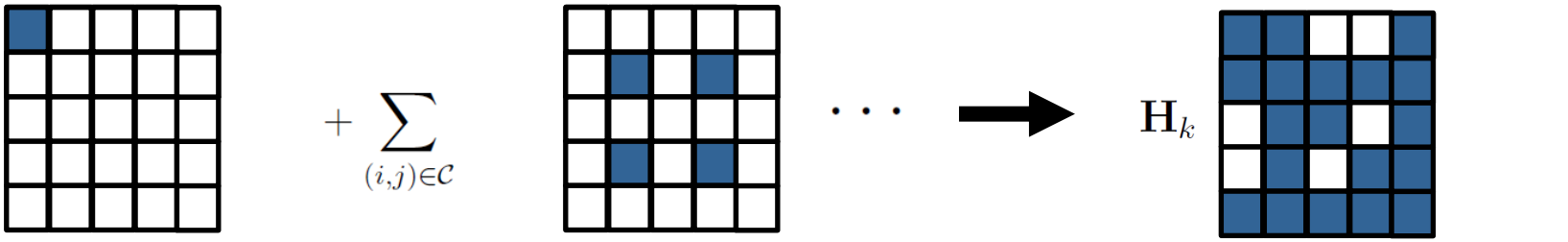




$$\mathbf{b}_k = \mathbf{J}_k^\top \mathbf{W}\mathbf{r}(\mathbf{x}) = \left(\mathbf{J}_k^0\right)^\top \boldsymbol{\Sigma}_{0,\boldsymbol{\xi}}^{-1}\mathbf{r}^0(\mathbf{x}_k) + \sum_{(i,j)\in\mathcal{C}} \left(\mathbf{J}_k^{i,j}\right)^\top \boldsymbol{\Sigma}_{i,j}^{-1}\mathbf{r}^{i,j}(\mathbf{x}_k)$$

Structure of the PGO Problem





Sparse, depending on constraint connectivity

$$\mathbf{H}_k = \mathbf{J}_k^\top \mathbf{W} \mathbf{J}_k = \left(\mathbf{J}_k^0\right)^\top \boldsymbol{\Sigma}_{0,\boldsymbol{\xi}}^{-1} \mathbf{J}_k^0 + \sum_{(i,j)\in\mathcal{C}} \left(\mathbf{J}_k^{i,j}\right)^\top \boldsymbol{\Sigma}_{i,j}^{-1} \mathbf{J}_k^{i,j}$$

Scale Consistency in Monocular SLAM

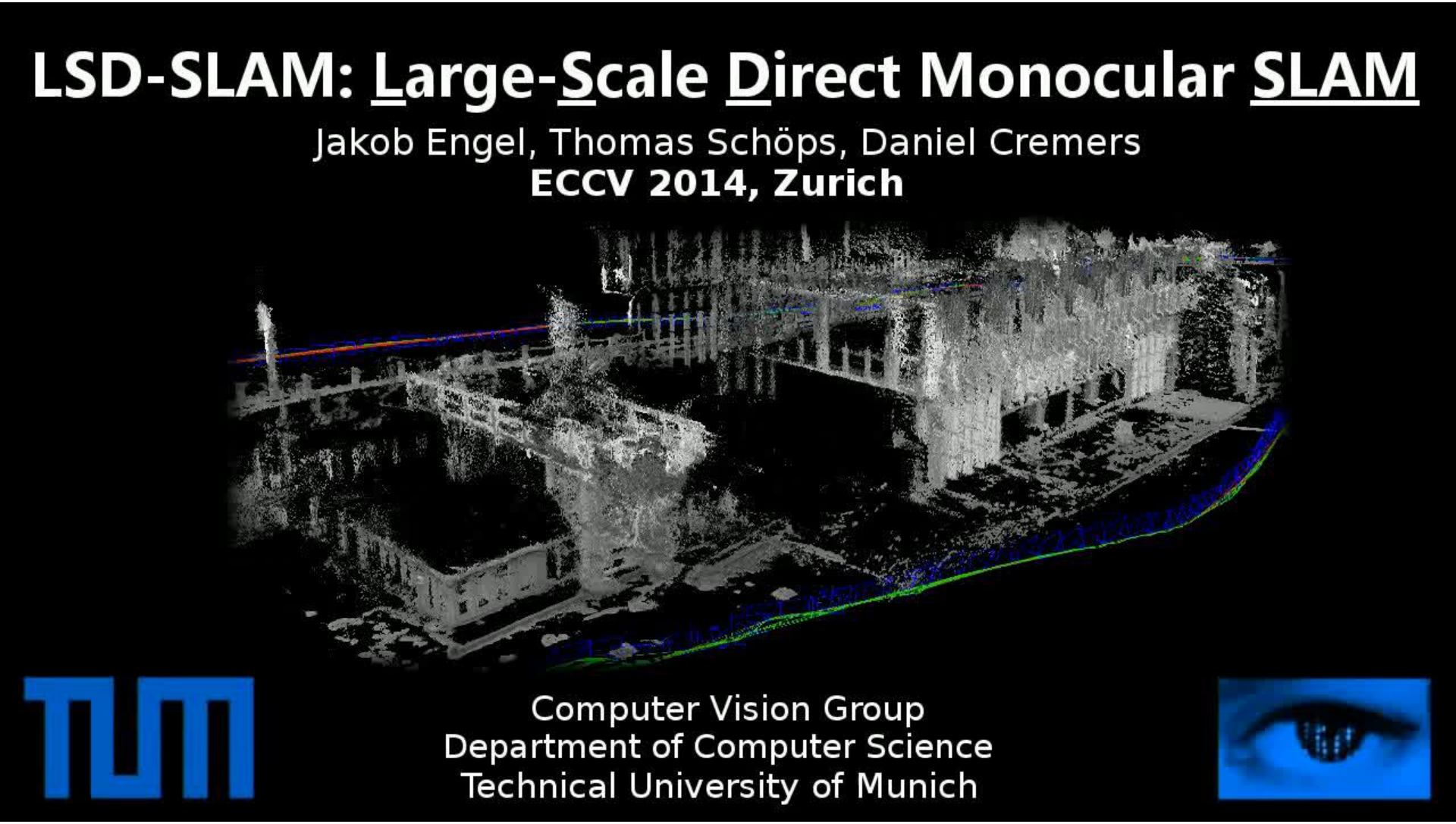
- Monocular SLAM: Scale not observable! Scale drifts!
  - Scale as an additional degree of freedom
  - Parametrize poses in group of similarity transformations $\mathbf{Sim}(3)$ instead of Euclidean transformations ($\mathbf{SE}(3)$)
  - Optimize for globally consistent scale

- Group of similarity transformations $\mathbf{Sim}(3)$
  - Group elements now include a scale parameter

$$\mathbf{T} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbf{Sim}(3)$$

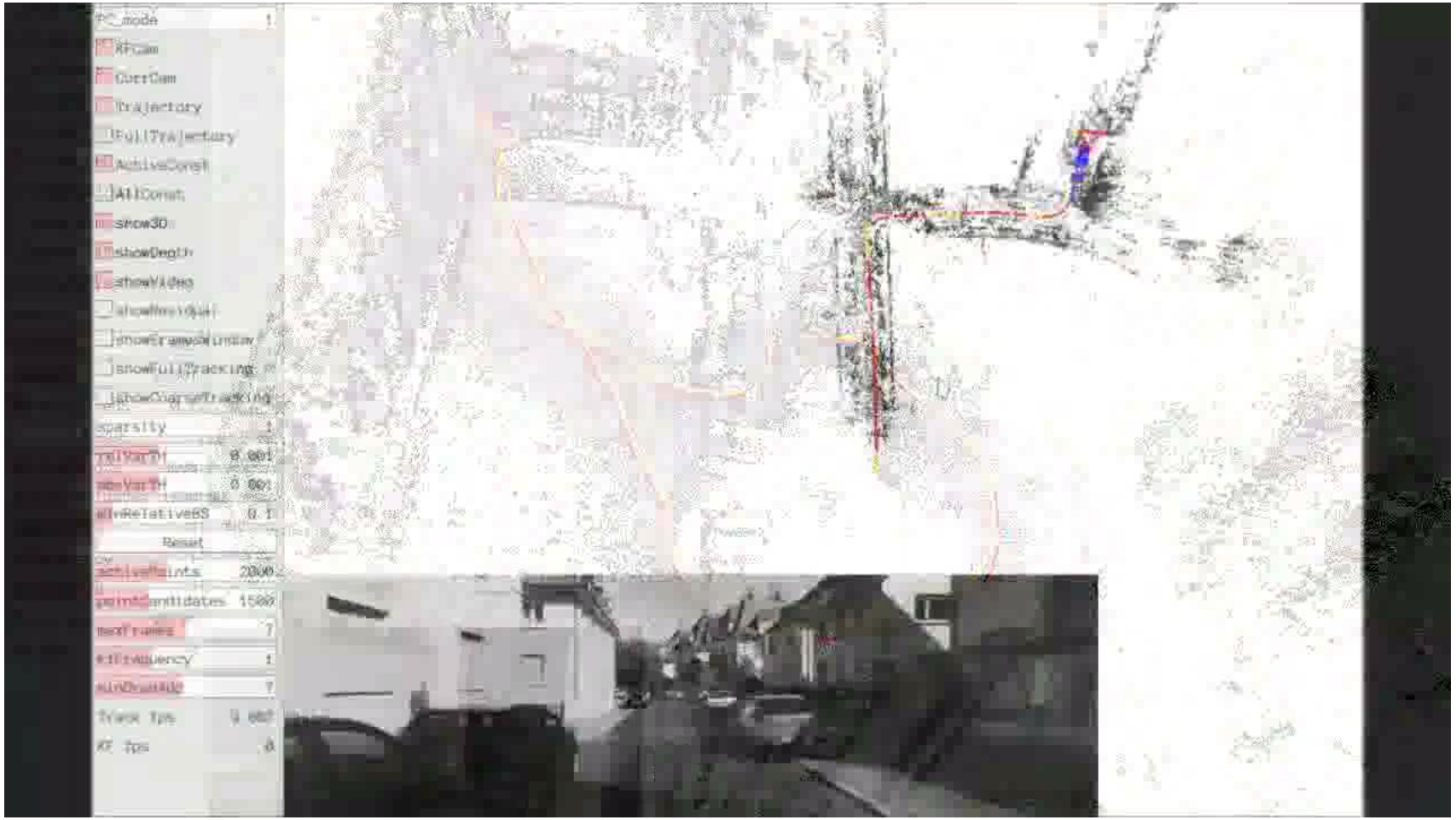
  - Also has an associated Lie algebra with exponential and logarithm map
  - Lie algebra elements have 7 degree of freedom, 6 for rigid motion, 1 for scale
  - See Strasdat et al., Scale Drift-Aware Large Scale Monocular SLAM, Robotics Science and Systems, 2010

Example: Scale Consistency in Mono SLAM



Engel et al., LSD-SLAM: Large-Scale Direct Monocular SLAM, ECCV 2014

Example: Scale Consistency in Mono SLAM

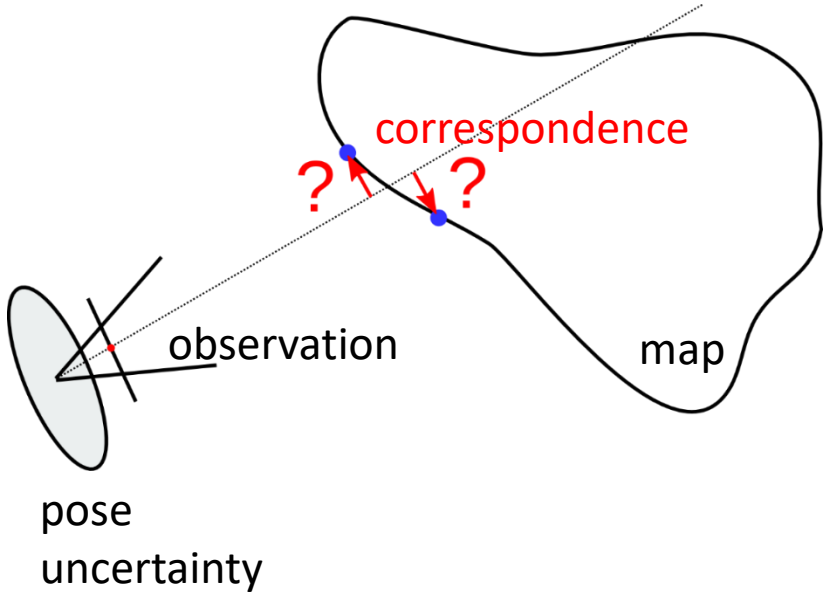


(Gao et al., LDSO, IROS 2018)

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

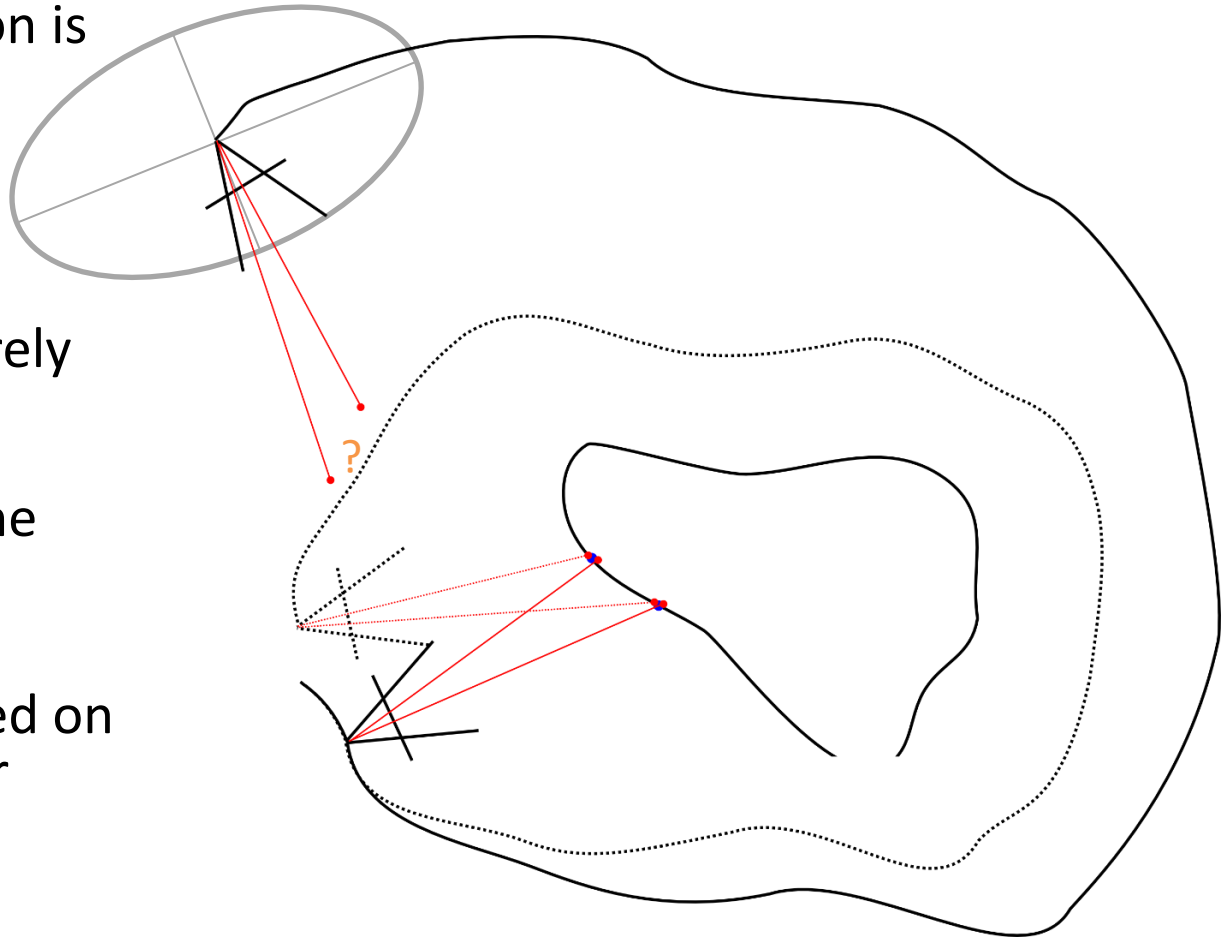
Short-Term Data Association Strategy In SLAM

- Similar to the data association problem in visual odometry
- Good initialization available due to image sequence
- E.g. interest point descriptors and RANSAC for robust association of point detections in the image with 3D point landmarks
  - Indirect approaches
- Or coarse-to-fine direct image alignment
  - Direct approaches



Loop Closure Detection

- Loop closure detection is a special case of data association
- Typically, we cannot rely on the state estimate because of the drift accumulated along the loop
- Data association based on cues such as shape or appearance needed (interest point descriptors, etc.)



Place Recognition

- Goal:
  - Find image correspondences between non-sequential frames
  - Detect when previous places are revisited

- Methods for detecting a revisit of previous places are often callsed “place recognition” in the SLAM literature

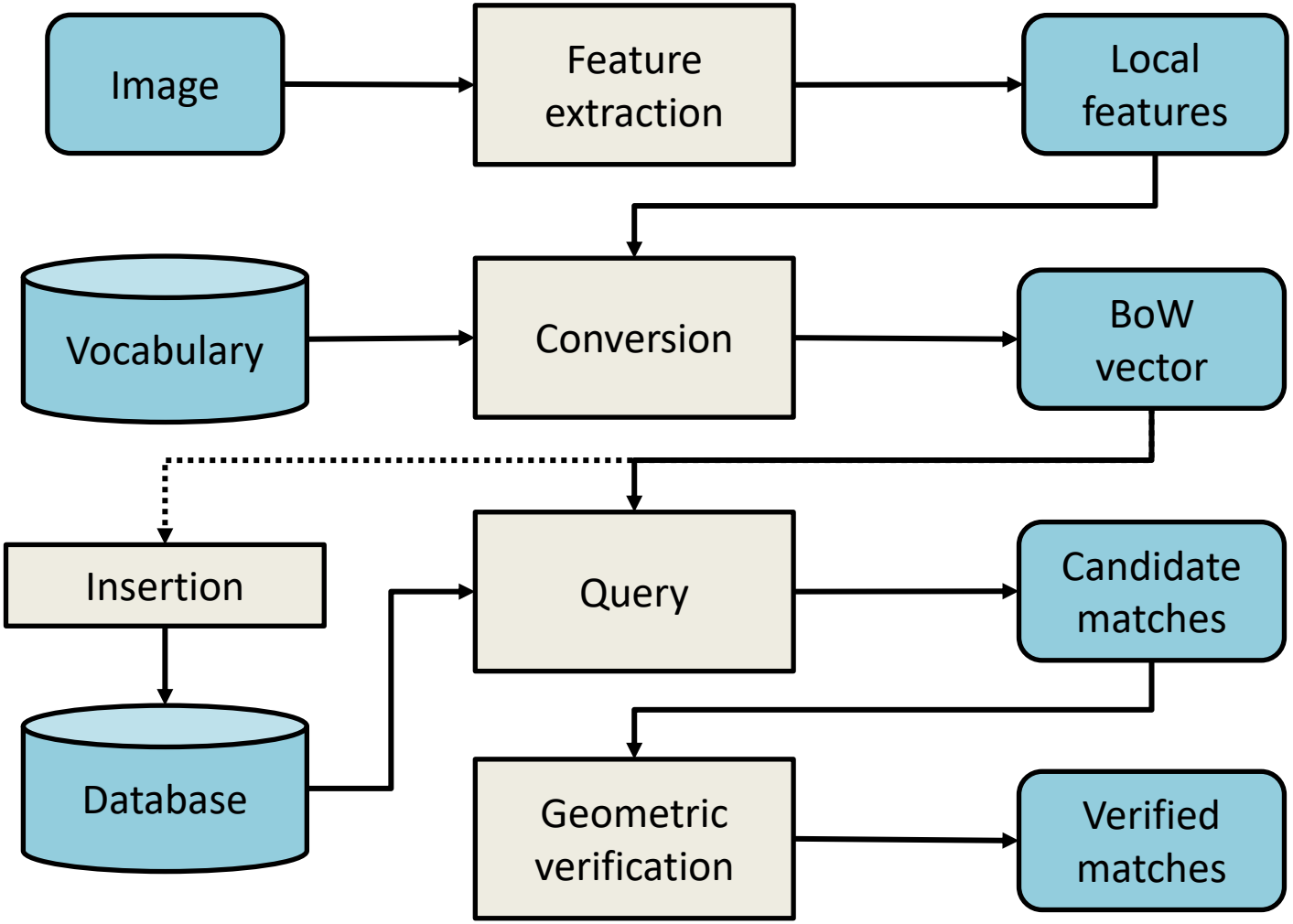
Images: Cummins and Newman, Highly Scalable Appearance-Only SLAM – FAB-MAP 2.0, RSS 2009

Place Recognition

- Idea: use image retrieval techniques
- Popular approach for place recognition is to use bag-of-visual-words based image retrieval
  - in conjunction with geometric verification
    - E.g. 8-point or PnP with RANSAC
- There are also deep learning based methods
  - However, they are in general not as runtime efficient
  - E.g. NetVLAD

Images: Cummins and Newman, Highly Scalable Appearance-Only SLAM – FAB-MAP 2.0, RSS 2009

Bag-of-Visual-Words based Image Retrieval

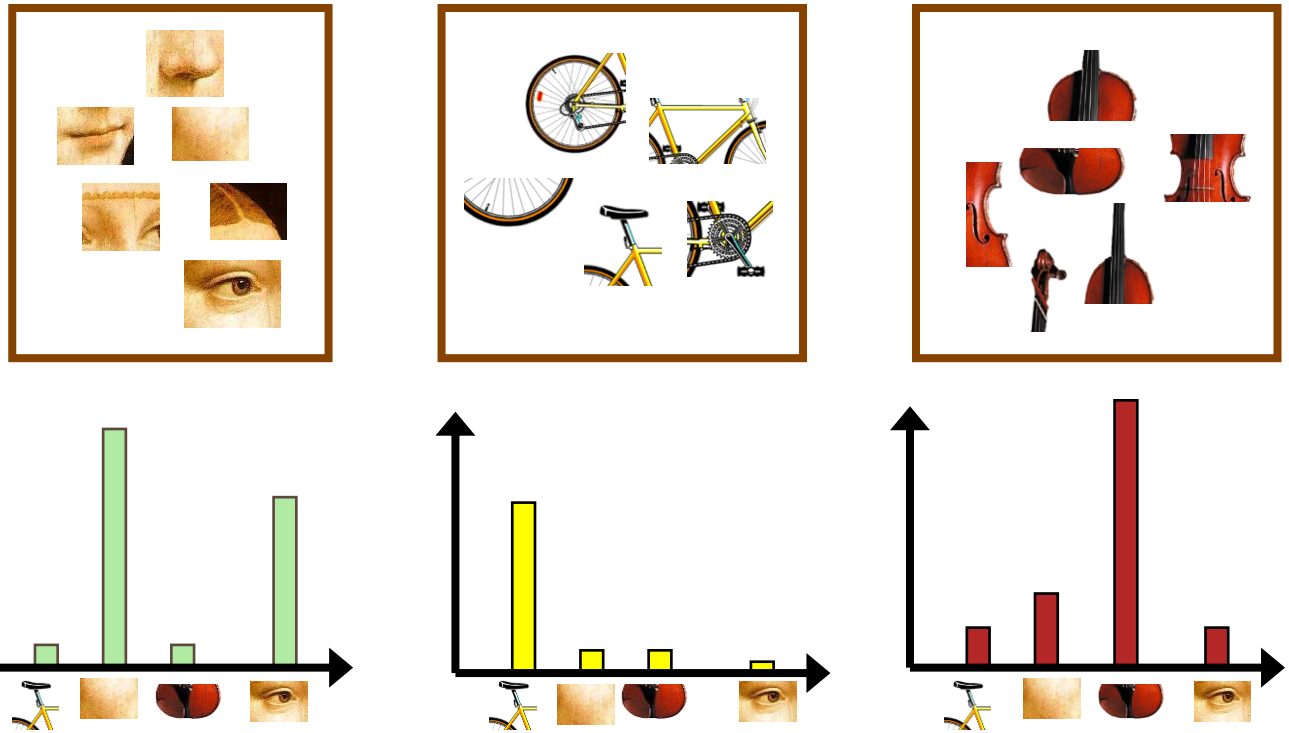


Bag of Visual Words

Slide credit: Svetlana Lazebnik

Bag of Visual Words

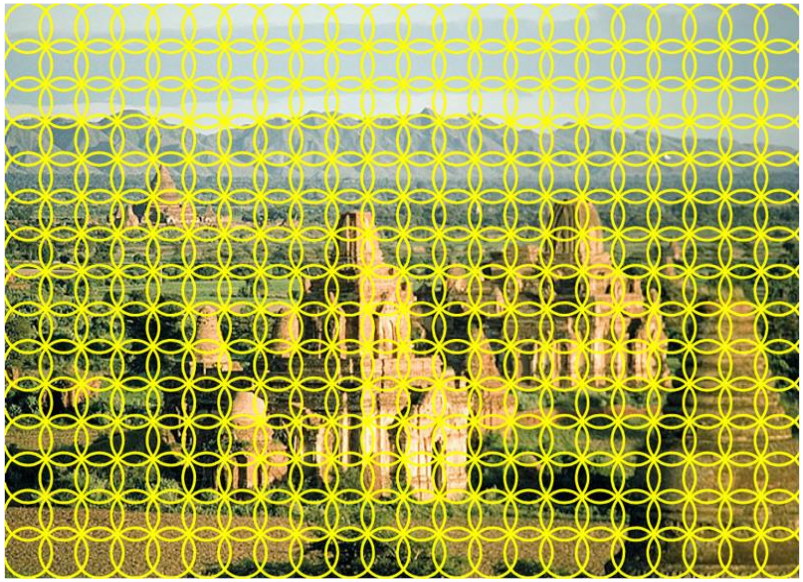
1. Extract local features
2. Learn “visual vocabulary”
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of “visual words”



Slide credit: Svetlana Lazebnik

Bag of Visual Words

1. Extract local features
   - in general using standard keypoint detectors + descriptors
   - Local features are supposed to be repeatable, invariant, distinct
   - Can be reused for pose refinement

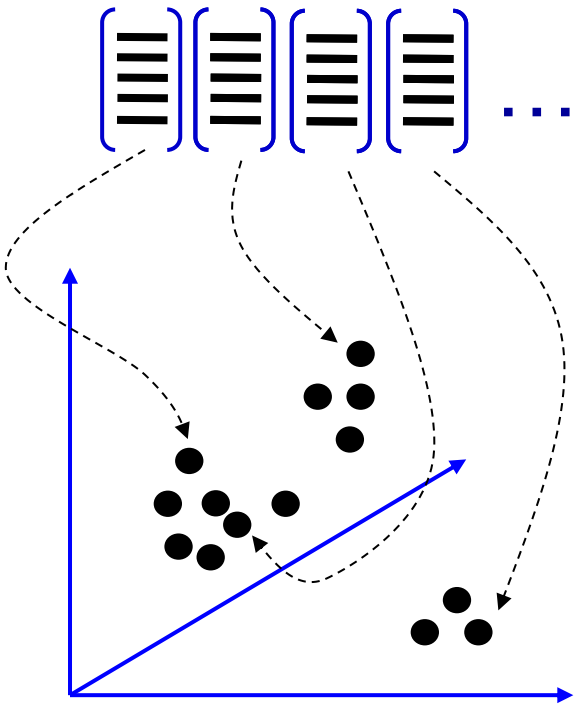


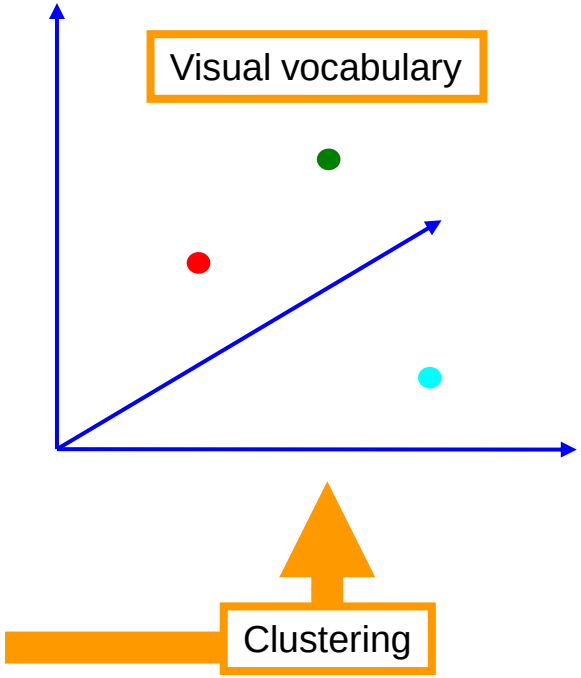
Slide credit: Svetlana Lazebnik

Bag of Visual Words

2. Learn “visual vocabulary”
   - Fixed size vocabulary learned based on descriptors obtained from a training dataset
   - E.g. using k-Means clustering

Extracted descriptors from the training set

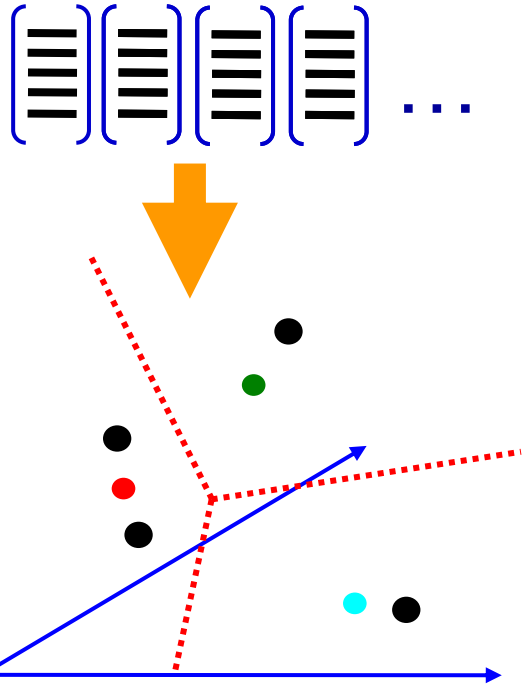




Slide credit: Svetlana Lazebnik, Josef Sivic

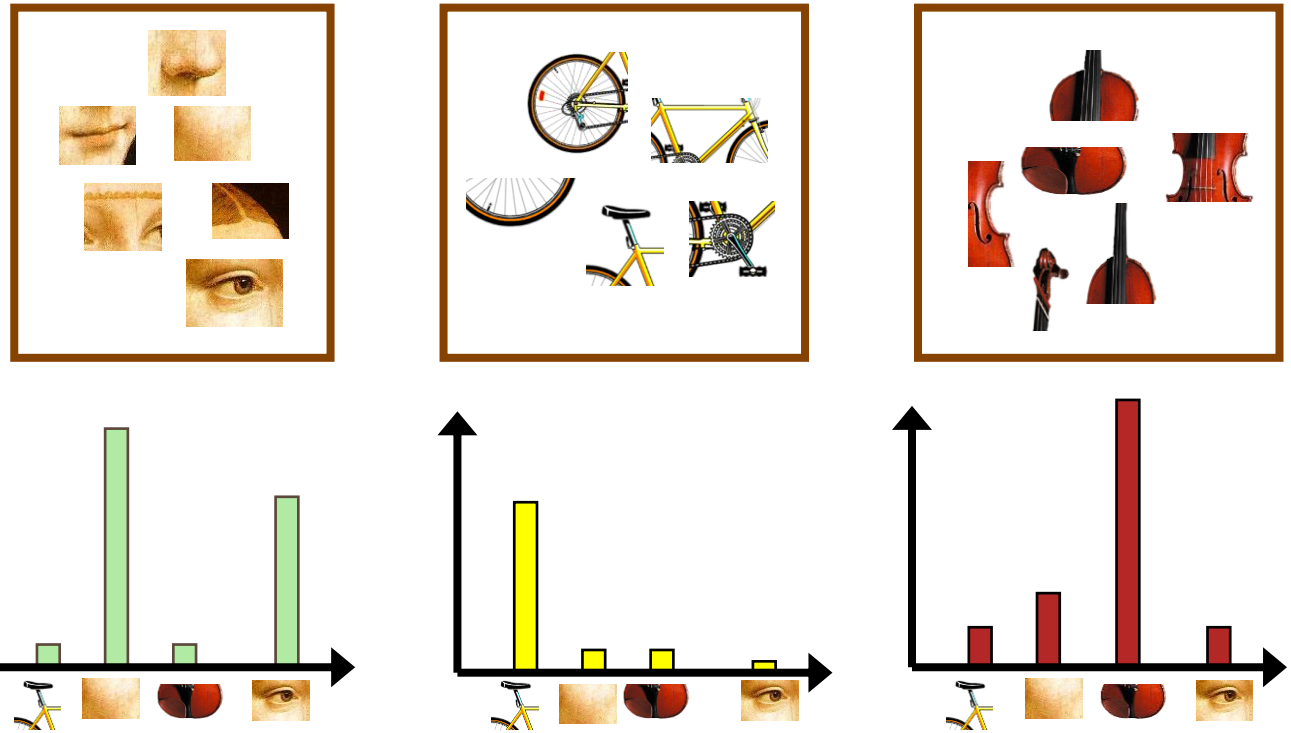
Bag of Visual Words

3. Quantize local features using visual vocabulary
   - For each feature find closest visual word in the vocabulary
   - Only need to keep the vocabulary index
   - It is important to have a well trained vocabulary
     - Cluster centers should represent descriptor distribution well



Bag of Visual Words

4. Represent images by frequencies of “visual words”
   - Calculate histogram of visual words (cluster indices) per image
     - How often does a certain word occur in the image
     - Term frequency (TF)



Slide credit: Svetlana Lazebnik, Josef Sivic

Bag of Visual Words

4. Represent images by frequencies of “visual words”

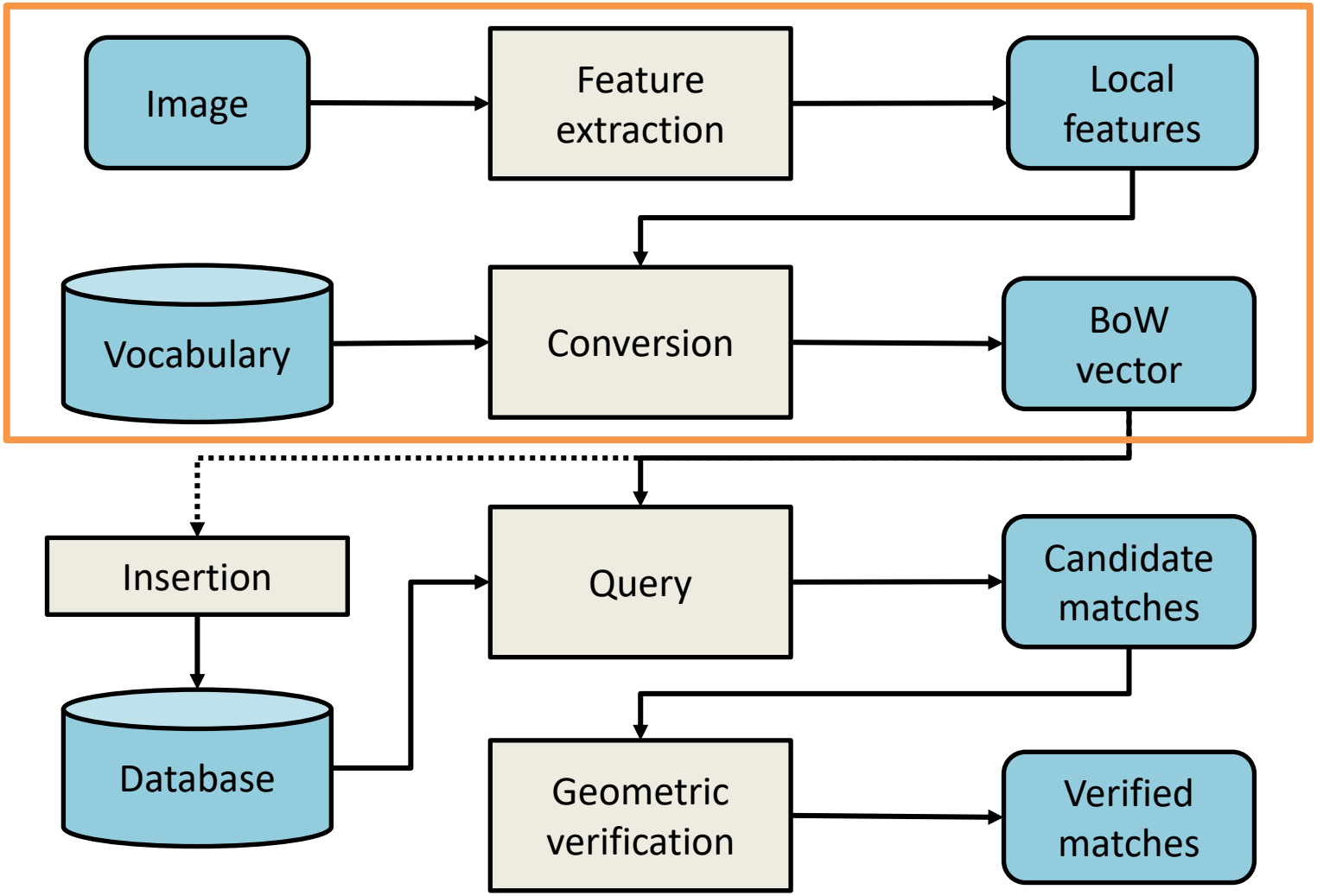
- Words occurring in every image are not very descriptive
  → downweight
  - Multiply TF by the inverse document frequency (IDF)

$$\text{IDF} = \log\left(\frac{\text{total number of image}}{\text{number of images containing word}}\right)$$

    - Since in SLAM the database is growing over time IDF has to be calculated from training dataset
  - TF-IDF = TF · IDF

- Normalize TF-IDF
  - Sum up to 1 over the entire vocabulary
  - BoW vector becomes independent of absolute number of features in an image

Bag-of-Visual-Words based Image Retrieval



Loop Closing is Difficult!

Image credit: Juan D. Tardós

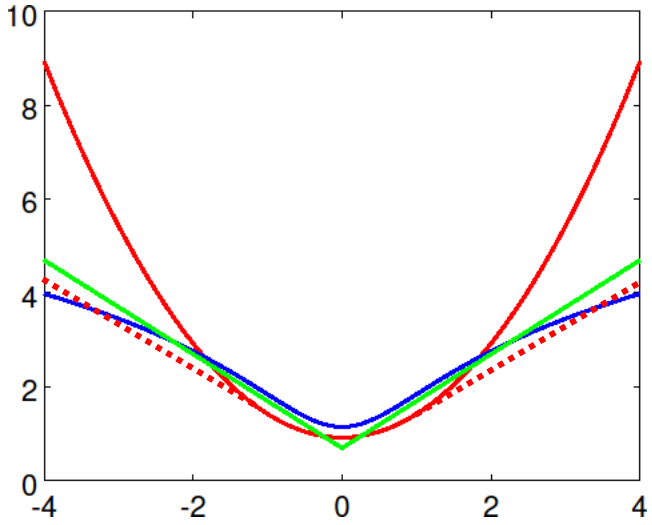
Robust Optimization

- Data association is hard
- Can we make SLAM optimization more robust to data association outliers?
- Gaussian noise assumption makes optimization sensitive to outliers
  - Use heavier-tail distributions / robust norms

Recap: Huber Loss

- Huber-loss „switches“ between Gaussian (locally at mean) and Laplace distribution

$$\|r\|_\delta = \begin{cases} 0.5r^2 & \text{for } |r| \leq \delta \\ \delta(|r| - 0.5\delta) & \text{otherwise} \end{cases}$$



- Normal distribution
- Laplace distribution
- Student-t distribution

Huber-loss for δ = 1

Outlook: Multi-Weather SLAM

- Idea: using place recognition to localize in a pre-build map

https://www.youtube.com/watch?v=PBAmpYwAY3g

Lessons Learned

- Pose graph optimization to approximate the full SLAM posterior with condensed relative pose measurements between frames

- Gauss-Newton approximation reveals the structure of pose graph optimization
  - Hessian is typically sparse, sparsity can be read directly from relative pose constraints in pose graph (edge structure)
  - Loop closures introduce correlations between non-sequential poses
  - Denser structure of Hessian limits efficiency, loop closures change structure significantly

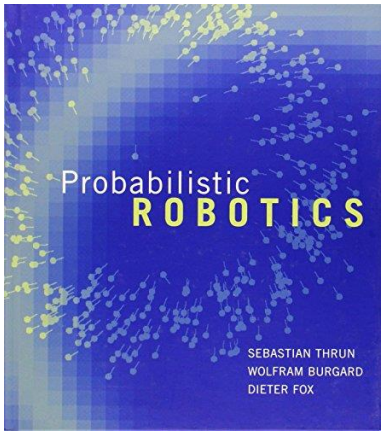
- Monocular SLAM using $\mathbf{Sim}(3)$ pose parametrization

Lessons Learned

- Loop closure detection through place recognition
- Place recognition by image retrieval techniques
  - Popular: Bag-of-Visual-Words + geometric verification (RANSAC)
- Increased robustness for data association outliers:
  - Heavier-tail residual distributions
  - E.g. Huber-Norm

Further Reading

- Probabilistic Robotics textbook



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

- Triggs et al., Bundle Adjustment – A modern Synthesis, Springer LNCS 1883, 2002
- Strasdat et al., Scale Drift-Aware Large Scale Monocular SLAM, Robotics Science and Systems, 2010
- R. Mur-Atal et al., ORB-SLAM: A Versatile and Accurate Monocular SLAM System, TRO 2015

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).