Robotic 3D Vision

Lecture 14: Visual SLAM 5 – DSO, VO/SLAM Summary

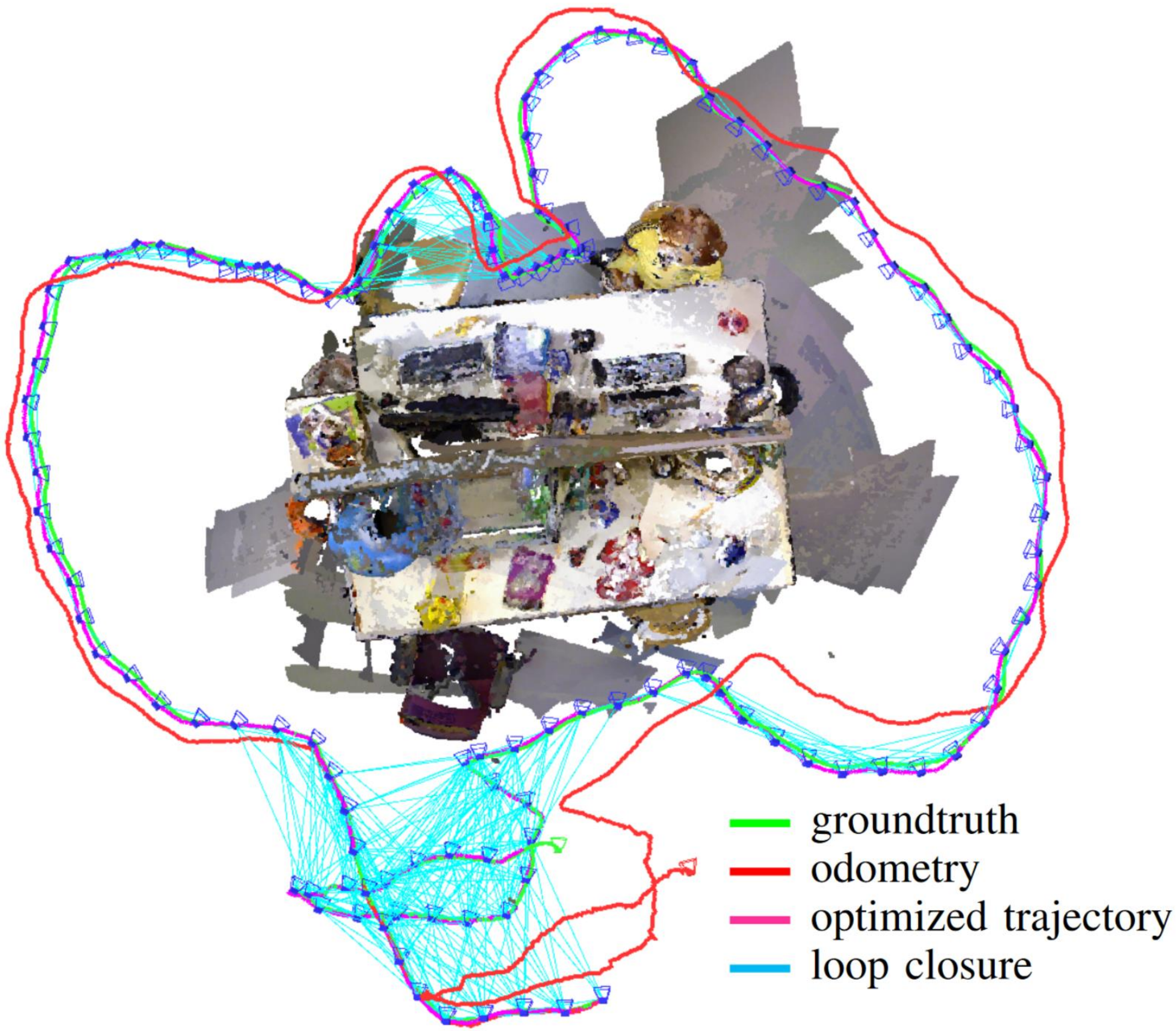
WS 2020/21

Dr. Niclas Zeller
Artisense GmbH

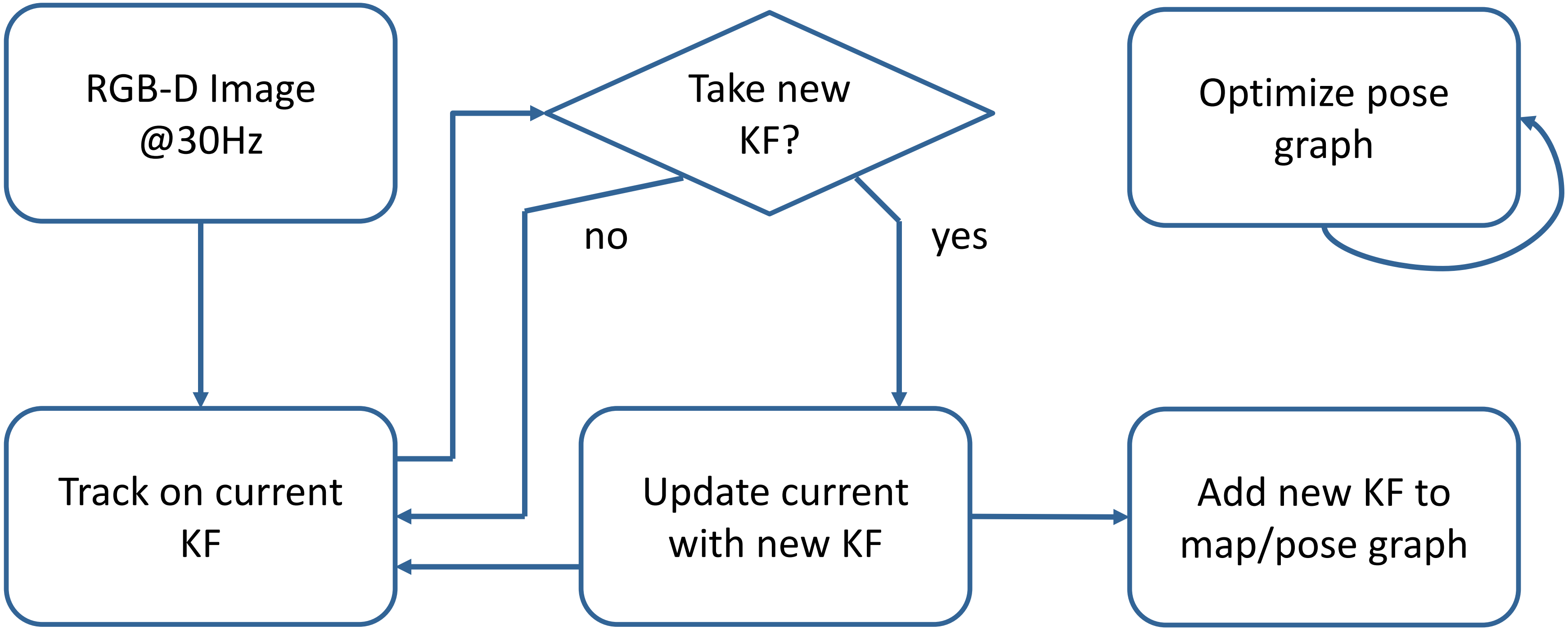
What We Will Cover Today

- Direct Sparse Odometry
- Summary on Visual Odometry and SLAM

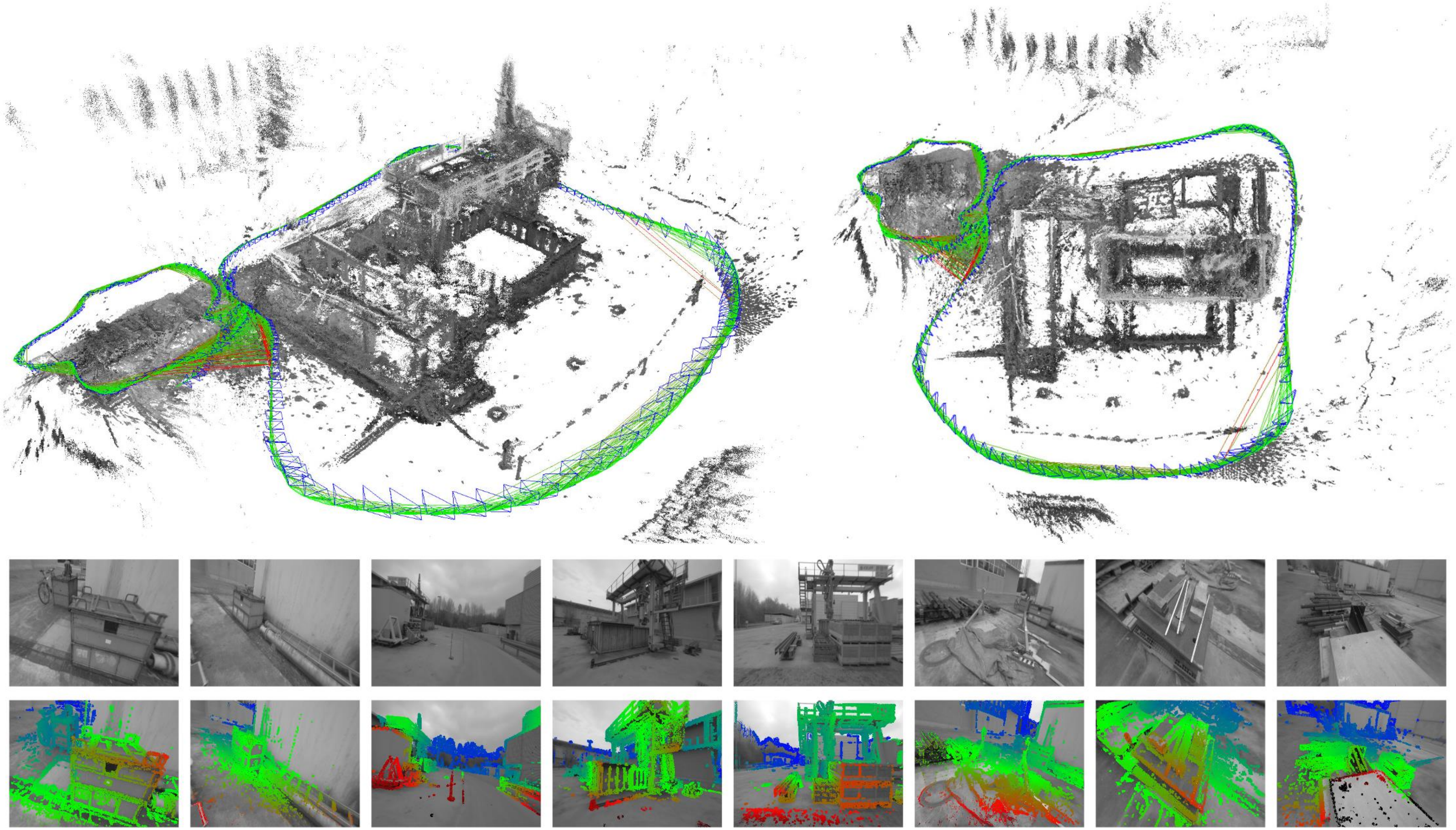
Recap: Direct SLAM with RGB-D Cameras



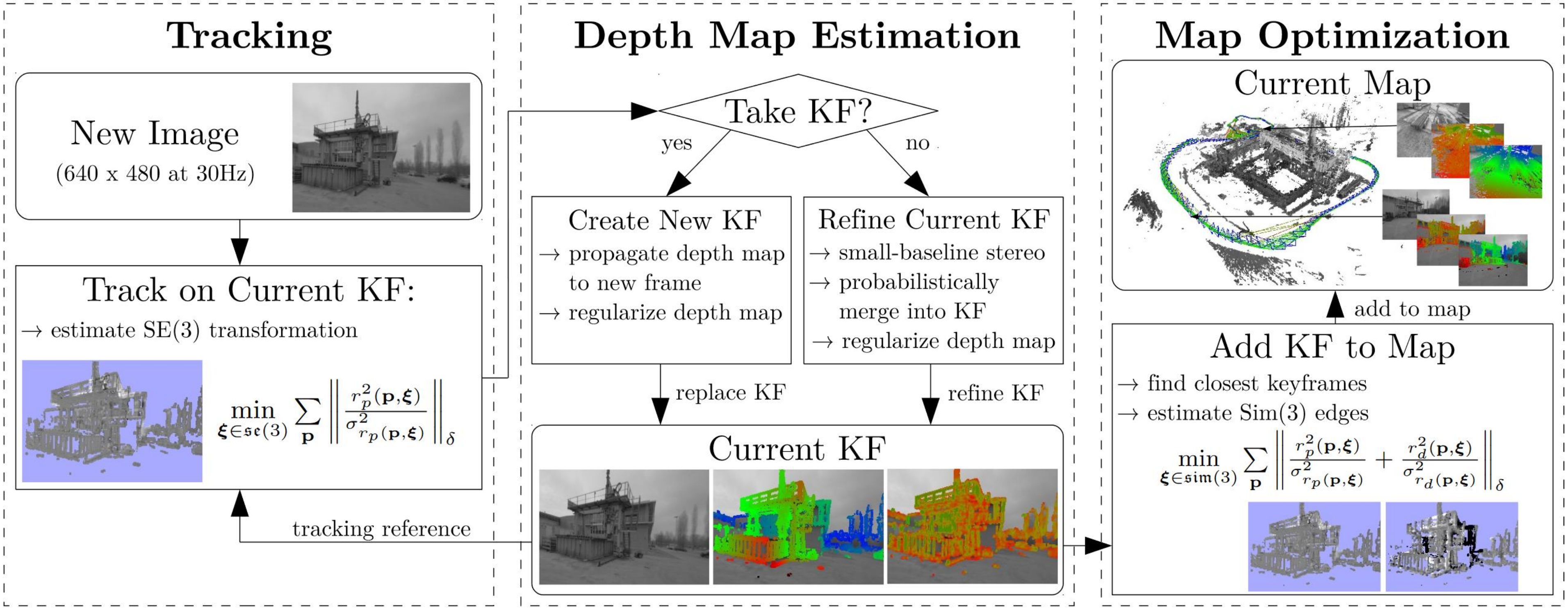
Recap: Algorithm Overview



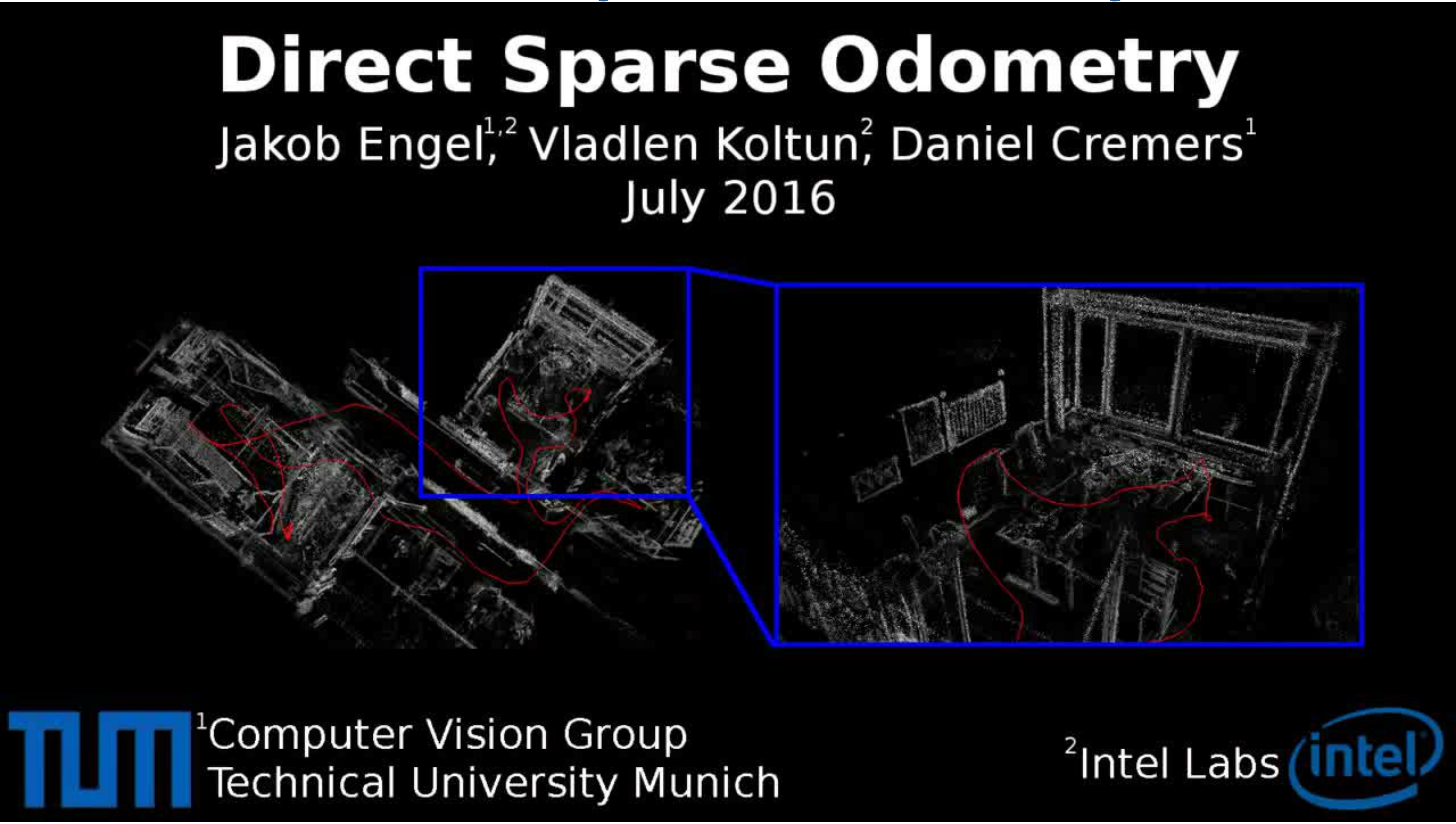
Recap: Large-Scale Direct Monocular SLAM



Recap: Algorithm Overview



Direct Sparse Odometry



Engel et al. T-PAMI 2018 https://www.youtube.com/watch?v=C6-xwSOOdqQ

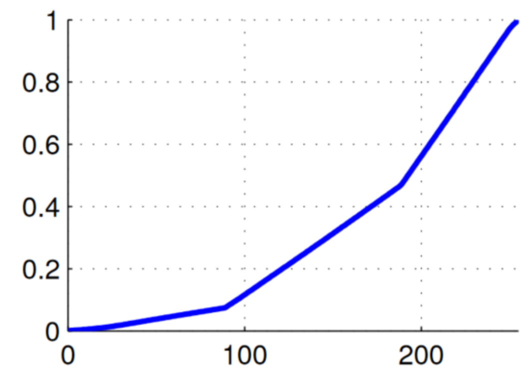
Recap: Camera Response Function

- The objects in the scene radiate light which is focused by the lens onto the image sensor

- The pixels of the sensor observe an irradiance $B : \Omega \rightarrow \mathbb{R}$ for an exposure time t

- The camera electronics translates the accumulated irradiance into intensity values according to a non-linear camera response function $G : \mathbb{R} \rightarrow [0, 255]$

example inv. G



- The measured intensity is $I(\mathbf{x}) = G(tB(\mathbf{x}))$

Recap: Vignetting

- Lenses gradually focus more light at the center of the image than at the image borders
- The image appears darker towards the borders
- Also called “lens attenuation”
- Lense vignetting can be modelled as a map $V : \Omega \to [0, 1]$

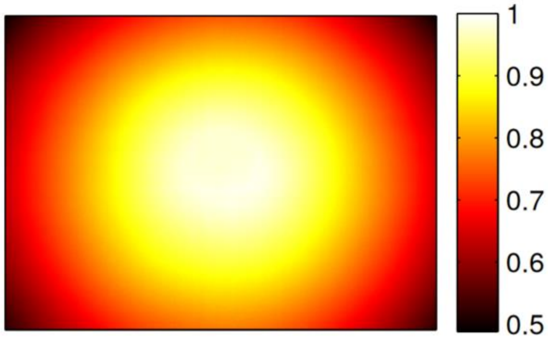
- Intensity measurement model

$$I(\mathbf{x}) = G(tV(\mathbf{x})B(\mathbf{x}))$$

uncorrected

corrected

$V(\mathbf{x})$

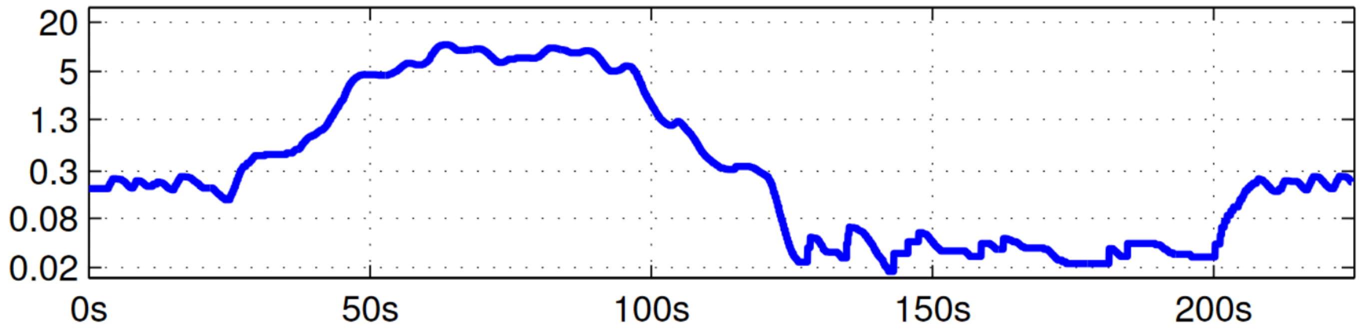


Recap: Brightness Constancy Assumption Revisited

- Camera images include vignetting effects and non-linear camera response function
- Idea: invert vignetting and camera response function using a known calibration
- Perform direct image alignment on irradiance images:

$$I'(\mathbf{y}) = tB(\mathbf{y}) = \frac{G^{-1}(I(\mathbf{y}))}{V(\mathbf{y})}$$

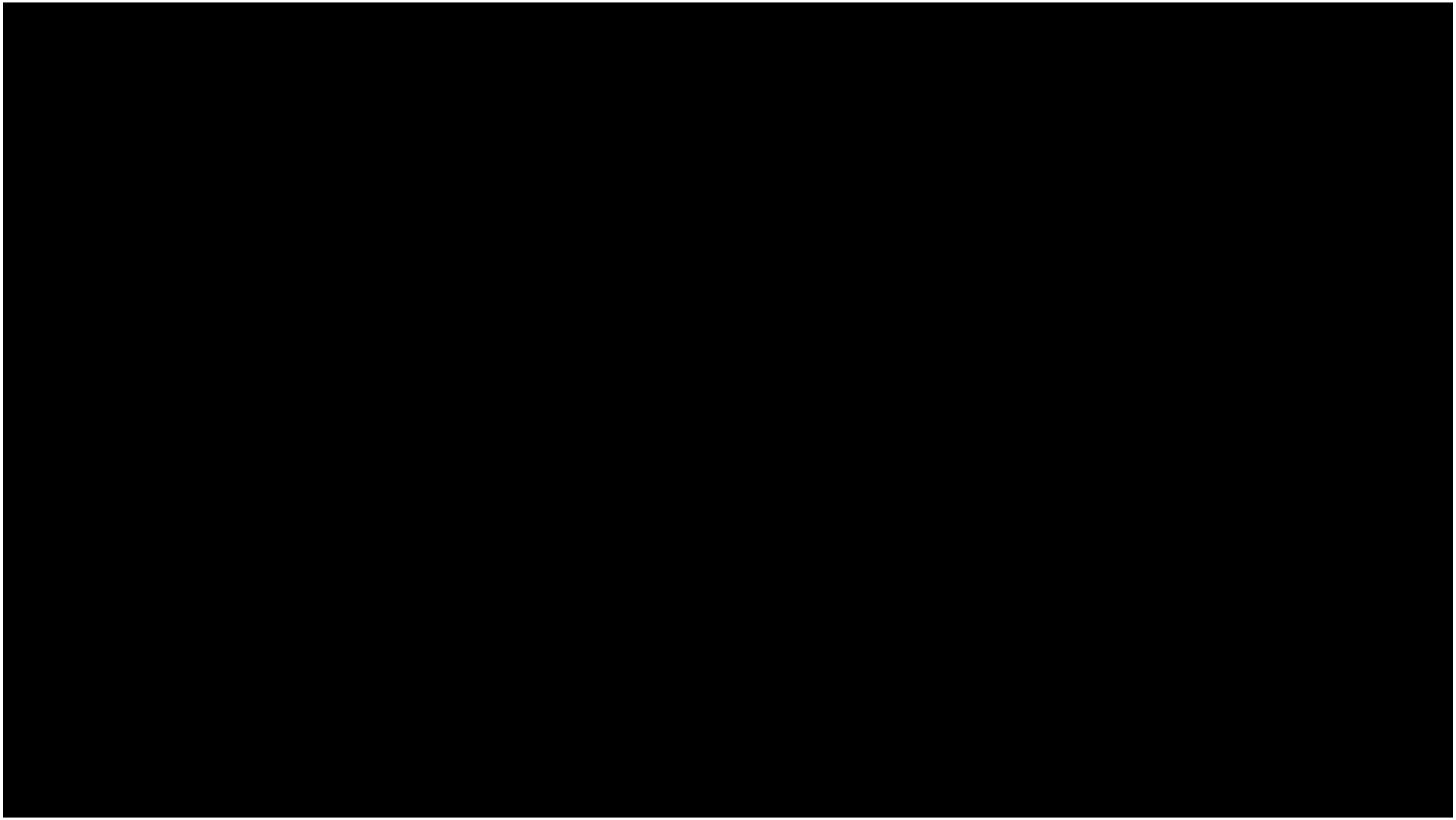
Recap: Brightness Constancy Assumption Revisited



- Automatic exposure adjustment needed in realistic environments
- Add affine exposure parameters explicitly to objective function:

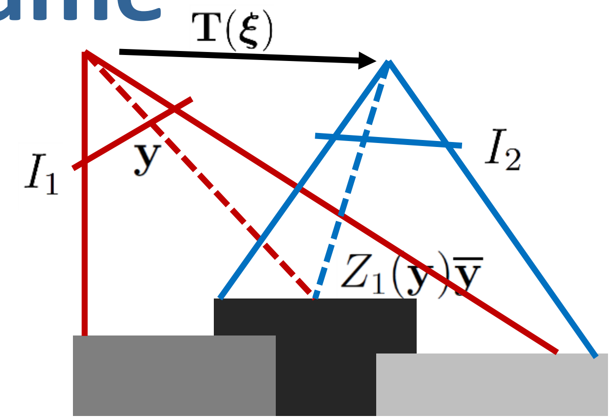
$$\left(I_2(\omega(\mathbf{y}, \boldsymbol{\xi}, Z_1(\mathbf{y}))) - b_2\right) - \frac{t_2 \exp(a_2)}{t_1 \exp(a_1)}\left(I_1(\mathbf{y}) - b_1\right)$$

Online Photometric Calibration



Bergmann et al., ICRA 2018 https://www.youtube.com/watch?v=nQHMG0c6Iew&feature=emb_logo

Tracking on Keyframe



- Direct image alignment of current frame to most recent keyframe

$$\zeta^* = \arg\min - \log(p(\zeta)) - \sum_{\mathbf{y}\in\Omega_Z} \log p(r(\mathbf{y},\zeta) \mid \zeta)$$

- Photometric residuals with affine paramters

$$r_I = \ (I_2(\omega(\mathbf{y},\boldsymbol{\xi},Z_1(\mathbf{y}))) - b_2) - \frac{t_2 \exp(a_2)}{t_1 \exp(a_1)}(I_1(\mathbf{y}) - b_1)$$

- Optimized parameters ζ now include affine parameters a_1, a_2, b_1, b_2
  - Can be set contatly to 0 if proper photometric calibration is available
- Exposure times t_1 and t_2 are set to 1 if not available

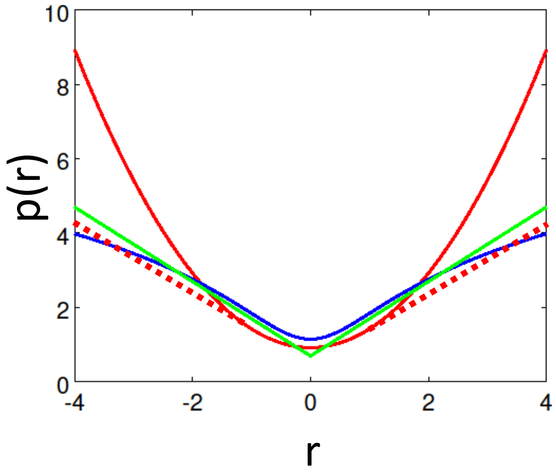
Tracking on Keyframe

- Residual distribution

$$E(\zeta) = \sum_{\mathbf{y} \in \Omega_Z} w_{\mathbf{y}} \, \|r(\mathbf{y}, \zeta)\|_\delta$$

  - Huber loss on residuals
  - Additional gradient dependent weight

$$w_{\mathbf{y}} := \frac{c^2}{c^2 + \|\nabla I_1(\mathbf{y})\|_2^2}$$



- Normal distribution
- Laplace distribution
- Student-t distribution

Huber-loss for $\delta = 1$

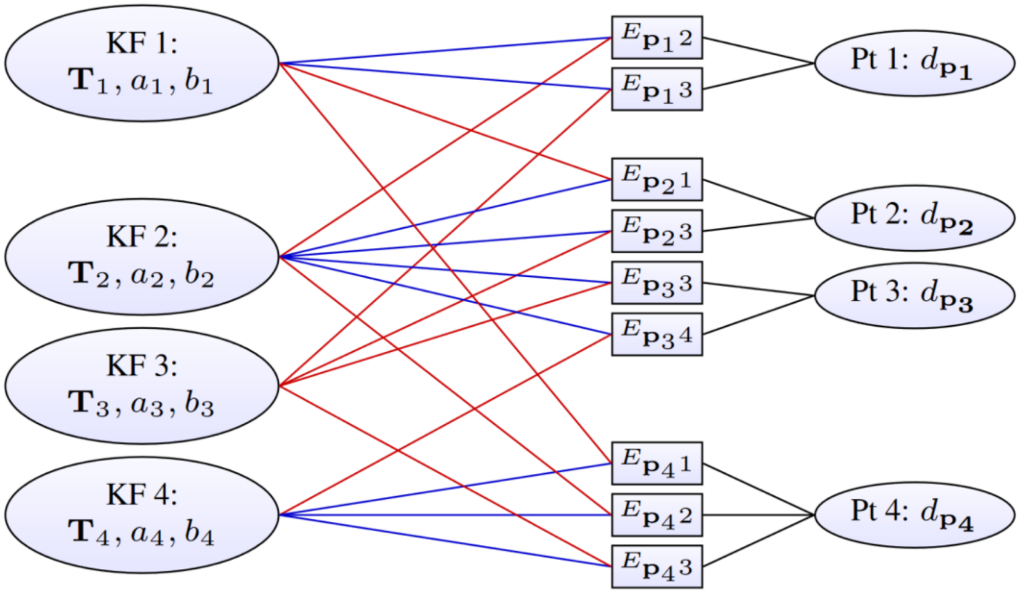
- Solved using iteratively reweighted least squares

Windowed Optimization

- Optimize in a recent window for
  - keyframe poses and photometric calibration
  - inverse depth of sparse set of active points
- Pose in SE(3)
- Marginalization of old variables

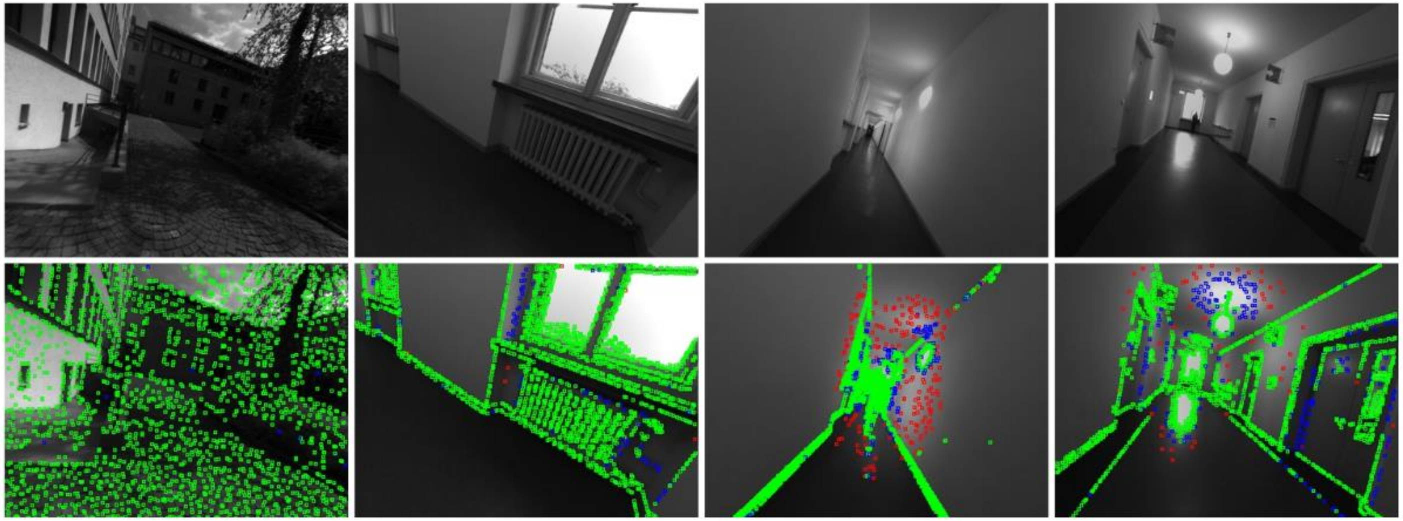
$$E_{\text{photo}} := \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \sum_{j \in \text{obs}(\mathbf{p})} E_{\mathbf{p}j}$$

$$E_{\mathbf{p}j} := \sum_{\mathbf{p} \in \mathcal{N}_{\mathbf{p}}} w_{\mathbf{p}} \left\| r(\mathbf{p}, \zeta_{ij}) \right\|_{\delta}$$



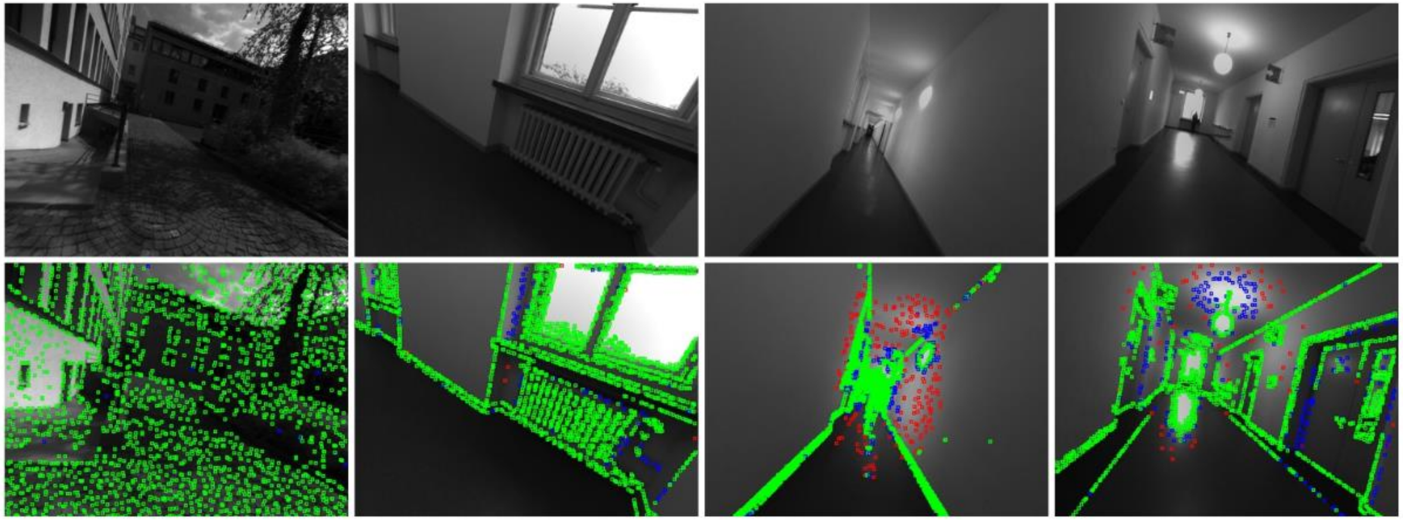
Depth Estimation

- Optimize inverse depth of a set of N_p points in all keyframes in bundle adjustment window
- Initialization of inverse depth of new points by fusion of short-baseline stereo comparisons from subsequent frames (similar to LSD-SLAM)



Depth Estimation

- Candidate point selection
  - Region-adaptive gradient threshold



Keyframe Selection

- Several criteria to decide when to create new keyframe
  - Mean square optical flow of points in latest keyframe towards current frame during tracking

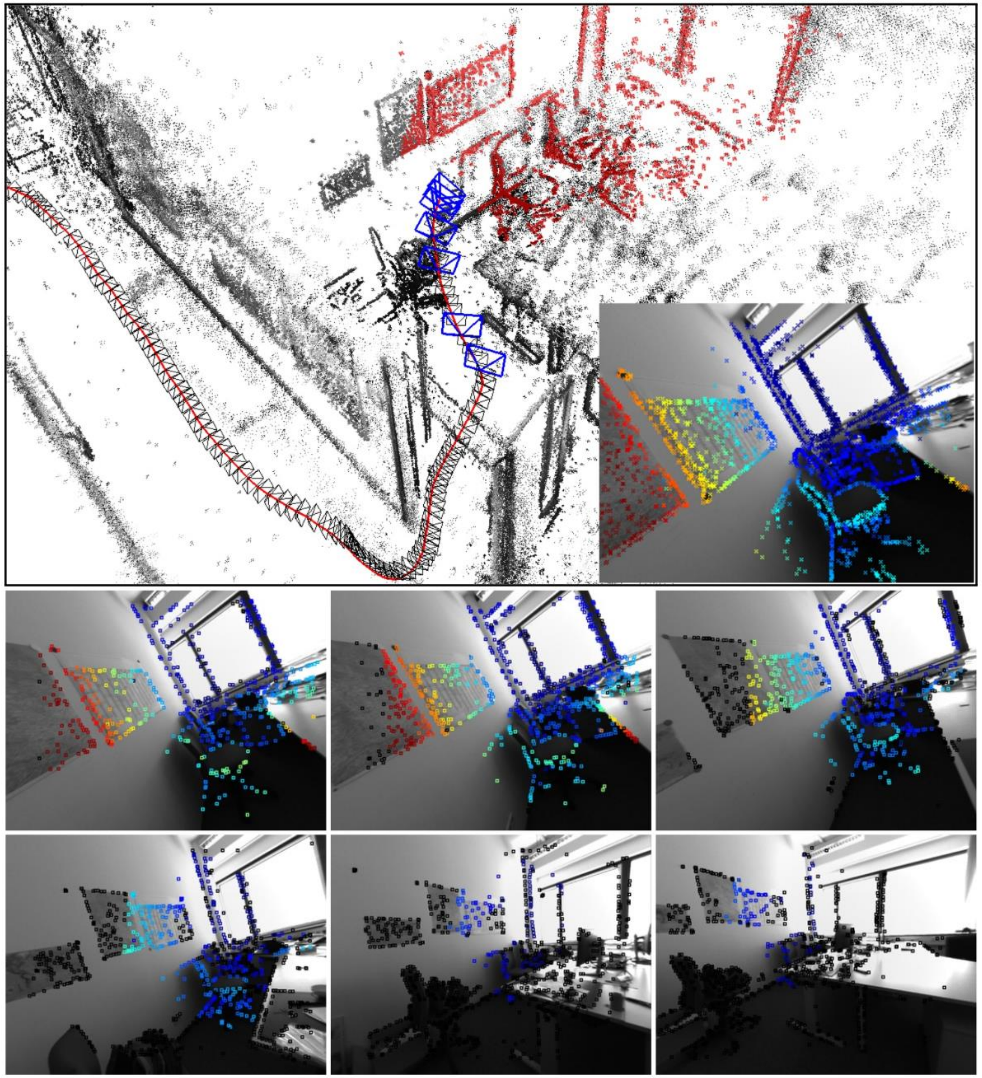
$$f := \left(\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{p} - \mathbf{p}'\|^2\right)^{\frac{1}{2}}$$

  - Relative brightness factor between keyframe and current frame

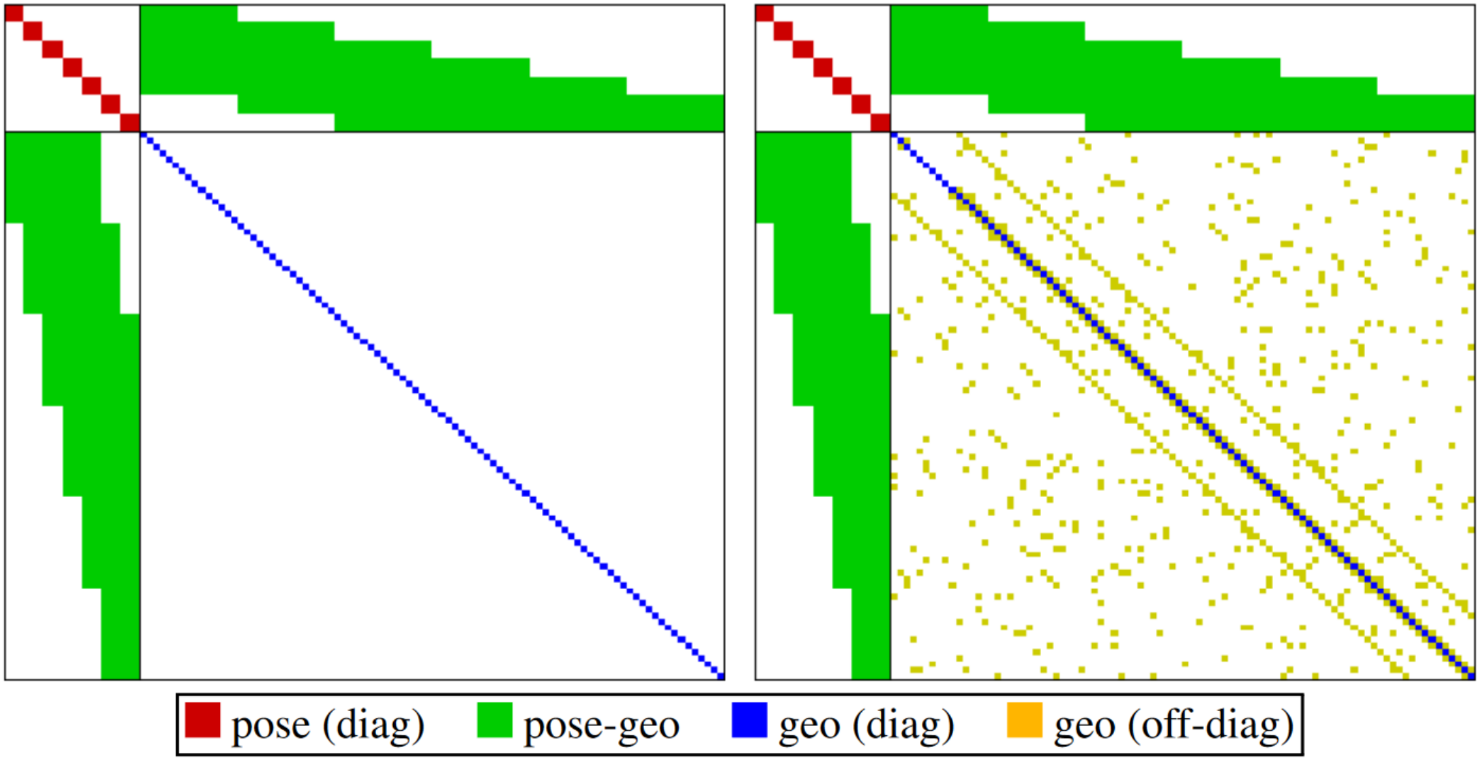
$$a := \left|\log(e^{a_j - a_i} t_j t_i^{-1})\right|$$

- Threshold linear combination of criteria
- Keyframes are generated with relatively high frequency

Keyframe Selection



Structure of the Hessian



- DSO neglects spatial correlations of depth estimates in image
- Hessian block on depths is diagonal

Marginalization

- Goal of marginalization is
  - to keep information of old poses and depths as prior without relinearizing and updating old variables
- Marginalization of a keyframe proceeds by
  - First marginalize all points hosted in the keyframe before the keyframe pose
  - Marginalize points without observations in last two keyframes
  - Drop observations of points from other keyframes in the marginalized keyframe to keep sparsity of Hessian

Recap: Gauss-Newton Method

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

$$\widetilde{E}(\mathbf{x}) = \frac{1}{2}\widetilde{\mathbf{r}}(\mathbf{x})^\top \mathbf{W}\widetilde{\mathbf{r}}(\mathbf{x})$$

$$= \frac{1}{2}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x} - \mathbf{x}_k\right)\right)^\top \mathbf{W}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x} - \mathbf{x}_k\right)\right) \qquad \mathbf{J}_k := \nabla_{\mathbf{x}}\mathbf{r}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$$

$$= \frac{1}{2}\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W}\mathbf{r}(\mathbf{x}_k) + \underbrace{\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W}\mathbf{J}_k}_{=:\mathbf{b}_k^\top}\left(\mathbf{x} - \mathbf{x}_k\right) + \frac{1}{2}\left(\mathbf{x} - \mathbf{x}_k\right)^\top \underbrace{\mathbf{J}_k^\top \mathbf{W}\mathbf{J}_k}_{=:\mathbf{H}_k}\left(\mathbf{x} - \mathbf{x}_k\right)$$

  - 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

Marginalization

- More formally, consider GN method for error function E(x)

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

- Split into variables $\mathbf{x}_\alpha$ to keep and $\mathbf{x}_\beta$ to marginalize

$$\begin{pmatrix} \mathbf{H}_{\alpha\alpha} & \mathbf{H}_{\alpha\beta} \\ \mathbf{H}_{\beta\alpha} & \mathbf{H}_{\beta\beta} \end{pmatrix} \begin{pmatrix} \mathbf{\Delta x}_\alpha \\ \mathbf{\Delta x}_\beta \end{pmatrix} = - \begin{pmatrix} \mathbf{b}_\alpha \\ \mathbf{b}_\beta \end{pmatrix}$$

- Applying the Schur complement yields

$$\widehat{\mathbf{H}}_{\alpha\alpha} = \mathbf{H}_{\alpha\alpha} - \mathbf{H}_{\alpha\beta} \mathbf{H}_{\beta\beta}^{-1} \mathbf{H}_{\beta\alpha}$$
$$\widehat{\mathbf{b}}_\alpha = \mathbf{b}_\alpha - \mathbf{H}_{\alpha\beta} \mathbf{H}_{\beta\beta}^{-1} \mathbf{b}_\beta$$

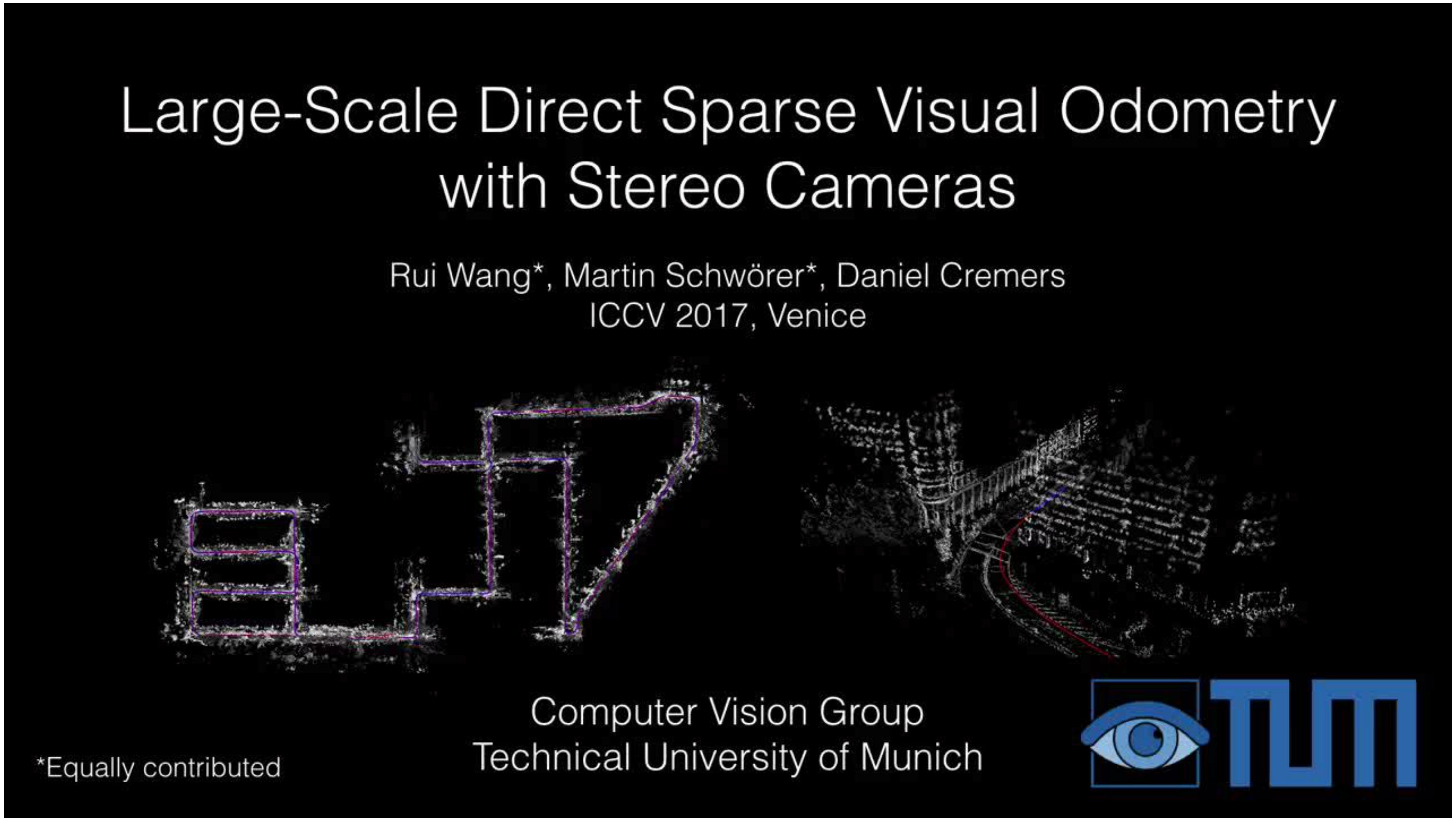
- Adds additional prior to GN optimization
- Sparsity of point Hessian is not affected by marginalization
  - Since corresponding aberservations are dropped

Marginalization

- Several criteria to decide when to marginalize a keyframe
  - Always keep the latest two keyframes
  - Keyframes with less than 5% visible points are marginalized
  - If more than N_f keyframes, marginalize keyframe which maximizes

$$s(I_i) = \sqrt{d(i,1)} \sum_{j \in [3,n] \setminus \{i\}} (d(i,j) + \epsilon)^{-1}$$

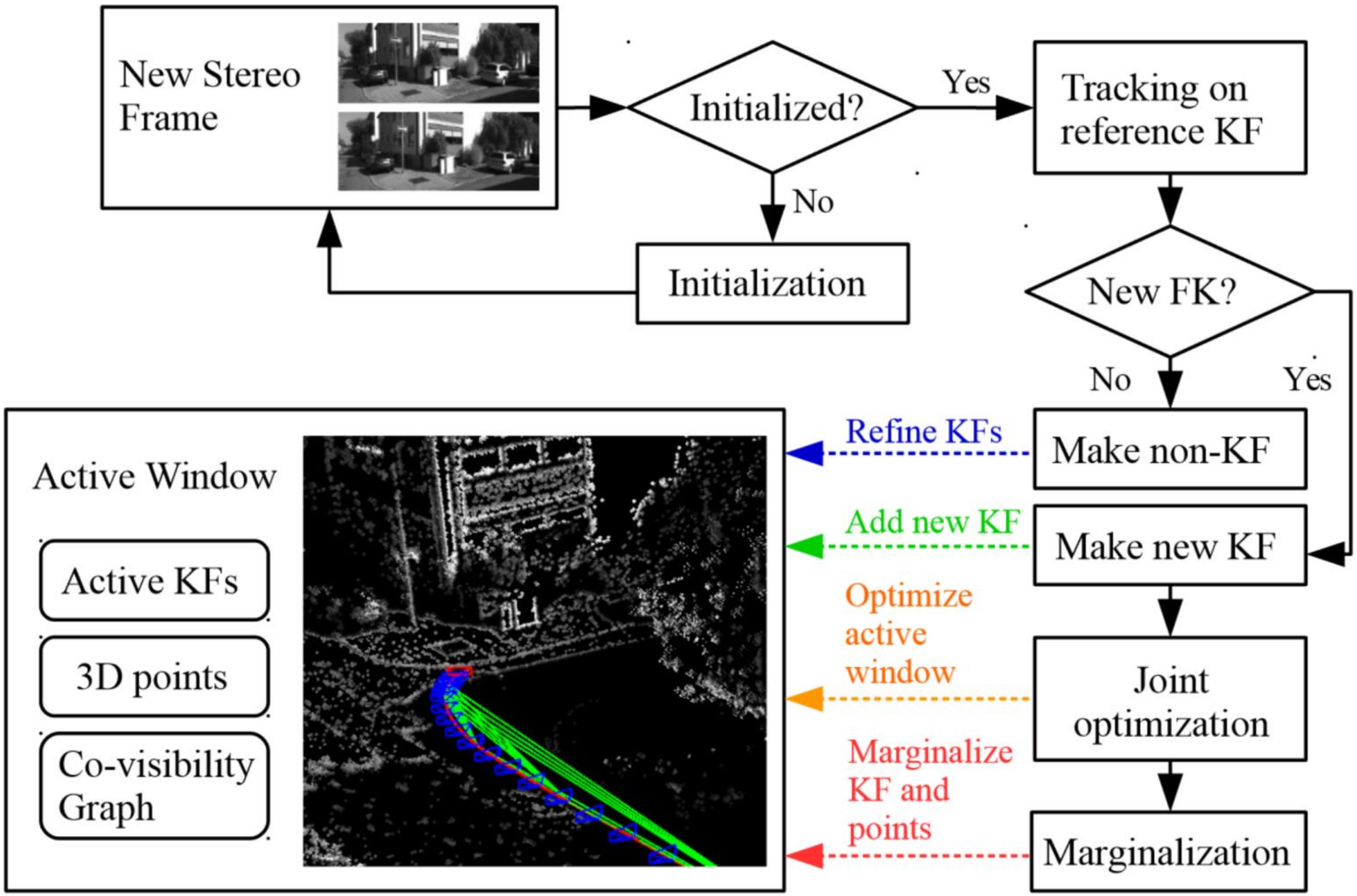
Stereo Direct Sparse Odometry



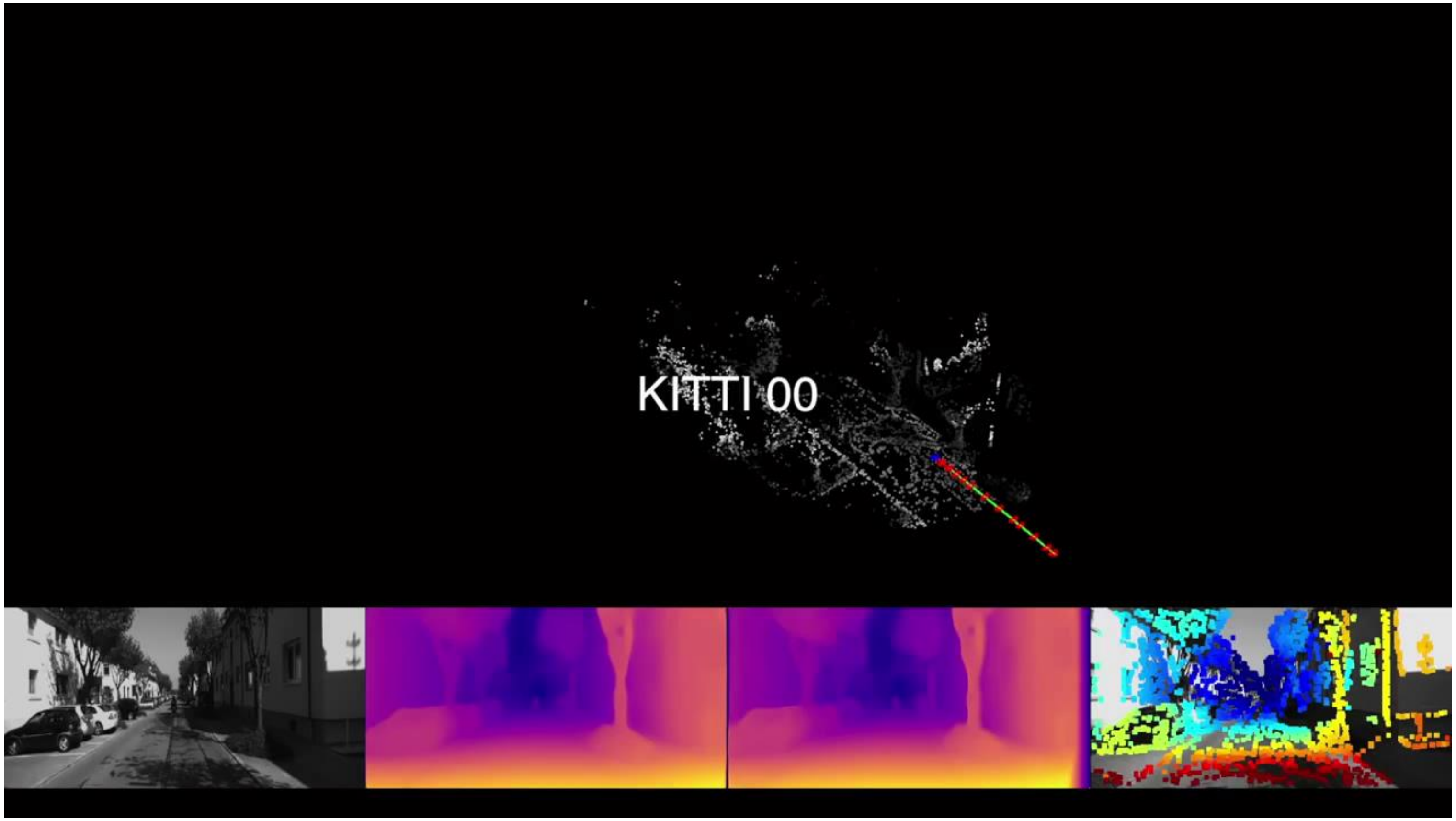
Wang et al. ICCV 2017

https://www.youtube.com/watch?v=A53vJO8eygw

Algorithm Overview



Deep Direct Sparse Odometry (Mono)



Yang et al. ECCV 2018

https://www.youtube.com/watch?v=sLZOeC9z_tw&t=7s

Comparison

DVO-SLAM	LSD-SLAM	DSO
+ RGB-D cameras	+ monocular cameras + stereo cameras	+ monocular cameras + stereo cameras
+ global consistency	+ global consistency	- no global consistency[1]
camera pose tracking towards keyframe	camera pose tracking towards keyframe	camera pose tracking towards keyframe
+ depth from sensor	+ depth from stereo comparisons & filtering	++ depth optimization using photometric residuals in local keyframe window
tracking-only & pose graph optimization	tracking-and-mapping & pose graph optimization	tracking-and-mapping & direct sparse bundle adjustment in local keyframe window with marginalization
+ local accuracy	+ local accuracy	++ local accuracy

[1]can be extended with PGO back-end (e.g. LDSO)

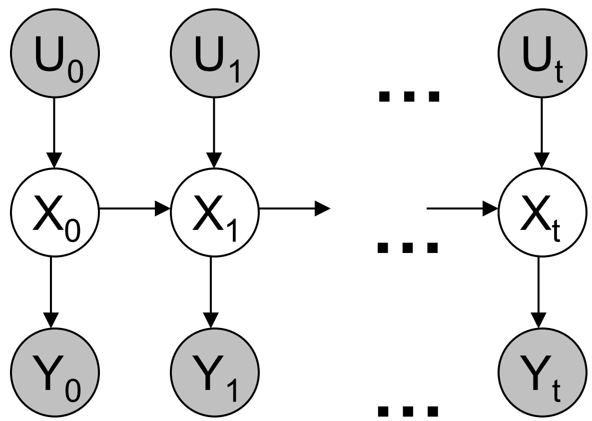
VO / VSLAM Summary

- Lecture blocks so far
  - Image formation and multiple view geometry
  - Probabilistic state estimation
  - Visual and visual-inertial odometry
  - Visual SLAM

- Outlook
  - 3D object detection and tracking
  - Dense reconstruction and map representations

Probabilistic State Estimation

- Probabilistic formulation of visual odometry and SLAM algorithms as inference in hidden Markov models



- Observation model $p\left(Y_t \middle| X_{0:t}, U_{0:t}, Y_{0:t-1}\right) = \boxed{p\left(Y_t \middle| X_t\right)}$

- State-transition model $p\left(X_t \middle| X_{0:t-1}, U_{0:t}\right) = \boxed{p\left(X_t \middle| X_{t-1}, U_t\right)}$

Probabilistic State Estimation

- Filtering: recursive estimation of most recent state (f.e. most recent camera pose)
  - Recursive Bayesian filter
  - (Extended) Kalman filter
  - Particle filter

Predict: $$p\left(X_t \mid y_{0:t-1}, u_{0:t}\right) = \int p\left(X_t \mid X_{t-1}, u_t\right) p\left(X_{t-1} \mid y_{0:t-1}, u_{0:t-1}\right) dX_{t-1}$$

observation y_t

action u_t

Correct: $$p\left(X_t \mid y_0, \ldots, y_t\right) = \frac{p\left(y_t \mid X_t\right) p\left(X_t \mid y_{0:t-1}, u_{0:t}\right)}{\int p\left(y_t \mid X_t\right) p\left(X_t \mid y_{0:t-1}, u_{0:t}\right) dX_t}$$

Probabilistic State Estimation

- Full state posterior estimation
  - Gaussian noise models, non-linear models leads to non-linear least squares
  - Gauss-Newton method, typically offline
  - Other noise models: Iteratively reweighted least squares

$$p\left(X_{0:t} \middle| U_{1:t}, Y_{0:t}\right) = p\left(X_0\right)\left(\prod_{\tau=0}^{t} \eta_\tau p\left(Y_\tau \middle| X_\tau\right)\right)\left(\prod_{\tau=1}^{t} p\left(X_\tau \middle| X_{\tau-1}, U_\tau\right)\right)$$

$$\Downarrow$$

$$\arg\min_{\mathbf{x}} E(\mathbf{x}) = \frac{1}{2}\mathbf{r}(\mathbf{x})^\top \mathbf{W}\mathbf{r}(\mathbf{x})$$

Probabilistic State Estimation

- Fixed-lag smoothing:
  - Inference of a window of recent states
  - Marginalization of remaining states
  - Trade-off between recursive filtering (faster) and full state posterior estimation (more accurate)
  - Marginalization does not have to be in temporally consistent order
    - See DSO
    - Strictly speaking no fixed-lag

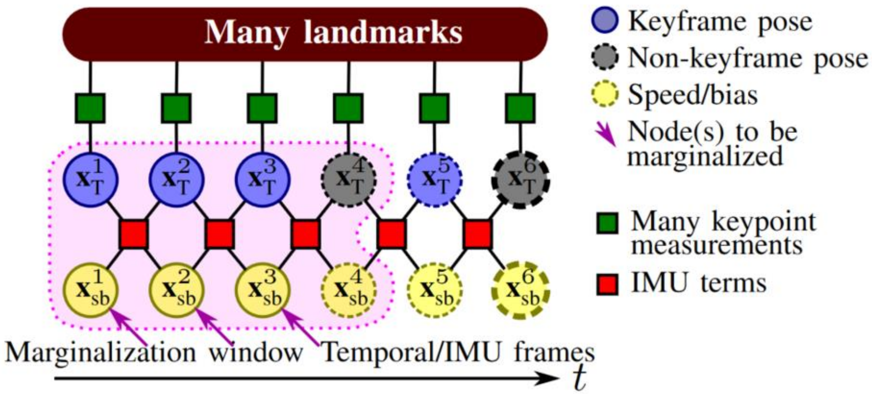


Image source: Leutenegger et al., IJRR 2015

State Estimation Approaches

Filtering	Fixed-Lag Smoothing	Maximum-A-Posteriori (MAP) Estimation
Recursive Bayesian filtering of the most recent state (e.g. Kalman Filter)	Optimize window of states through non-linear optimization and marginalization of old states	Full posterior optimization of all states through non-linear least squares
- Single linearization	+ Relinearize (in window)	+ Relinearize
- Accumulation of linearization errors	- Accumulation of linearization errors	+ Sparse Matrices
- Gaussian approximation of marginalized states	- Gaussian approximation of marginalized states	+ No Gaussian approximation of states
+ Very Fast	+ Fast	+ Slow

Visual Odometry vs. SLAM

Visual Odometry	Visual SLAM
Estimate motion of object from measurements of visual sensor on the object	Estimation motion of object and map of environment from measurements of visual sensor on the object
Real-time tracking	Real-time tracking, lower frame-rate loop closing and global optimization
Local consistency, drift	Local and/or global consistency
Map/3D reconstruction as a side-product	Concurrent accurate map estimation/3D reconstruction

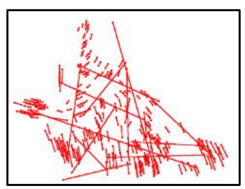
Indirect vs. Direct Methods

Indirect

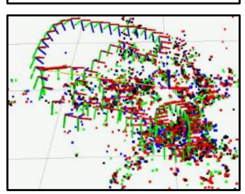
Input images

Extract and match keypoints (SIFT,BRIEF,…)

Track: min. reprojection error (point distances)



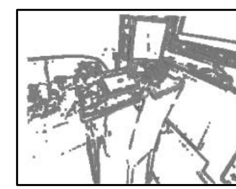
Map: estimate keypoint parameters (f.e. 3D coordinates)



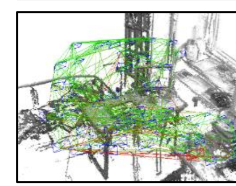
Direct

Input images

Track: min. photometric/ geometric error pixel-wise

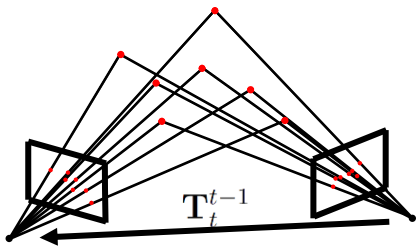


Map: estimate per-pixel depth from photoconsistency

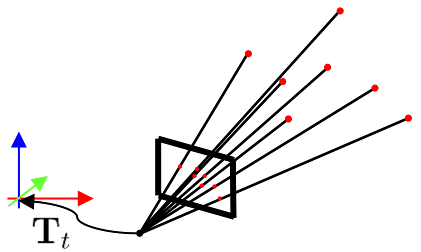


Motion Estimation from Point Correspondences

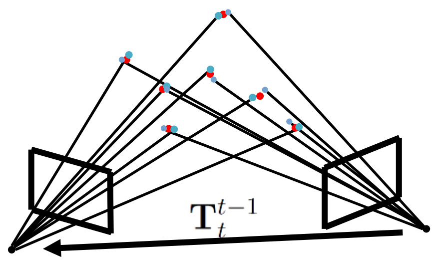
- **2D-to-2D**
  - Reproj. error:
$$E\left(\mathbf{T}_t^{t-1}, X\right) = \sum_{i=1}^{N} \|\bar{\mathbf{y}}_{t,i} - \pi\left(\bar{\mathbf{x}}_i\right)\|_2^2 + \left\|\bar{\mathbf{y}}_{t-1,i} - \pi\left(\mathbf{T}_t^{t-1}\bar{\mathbf{x}}_i\right)\right\|_2^2$$
  - Linear algorithm: 8-point



- **2D-to-3D**
  - Reprojection error: $E(\mathbf{T}_t) = \sum_{i=1}^{N} \|\mathbf{y}_{t,i} - \pi(\mathbf{T}_t\bar{\mathbf{x}}_i)\|_2^2$
  - Linear algorithm: DLT PnP



- **3D-to-3D**
  - Reprojection error: $E\left(\mathbf{T}_t^{t-1}\right) = \sum_{i=1}^{N} \left\|\bar{\mathbf{x}}_{t-1,i} - \mathbf{T}_t^{t-1}\bar{\mathbf{x}}_{t,i}\right\|_2^2$
  - Linear algorithm: Arun‘s method



Motion Estimation for Camera Type

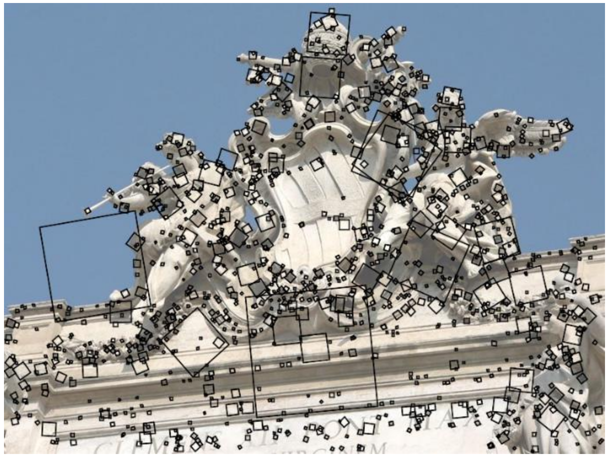
Correspondences	Monocular	Stereo	RGB-D
2D-to-2D	X	X	X
2D-to-3D	X	X	X
3D-to-3D		X	X

Keypoint Detection

- Desirable properties of keypoint detectors for visual odometry:
  - high repeatability,
  - localization accuracy,
  - robustness,
  - invariance,
  - computational efficiency

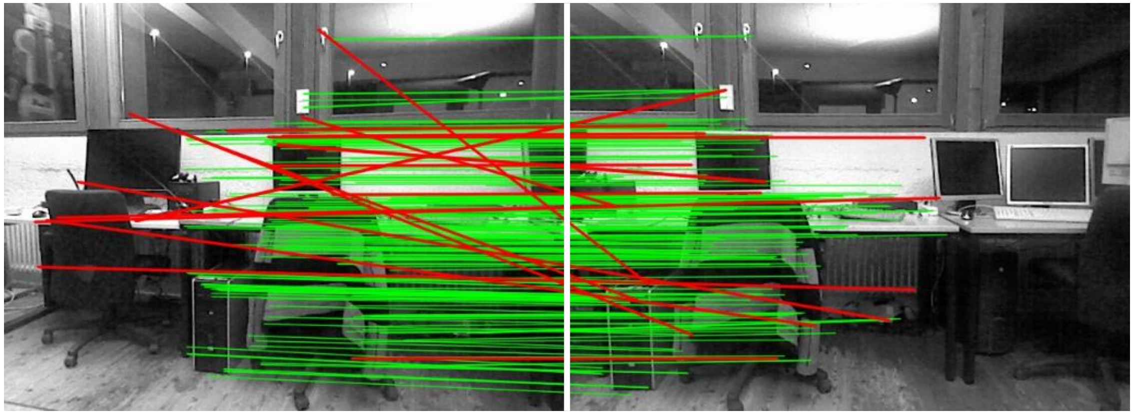
Harris Corners

Image source: Svetlana Lazebnik



DoG (SIFT) Blobs

Keypoint Matching



- Desirable properties for VO:
  - High recall
  - Precision
  - Robustness
  - Computational efficiency
- One possible approach to keypoint matching: by descriptor
- Robustness: RANSAC

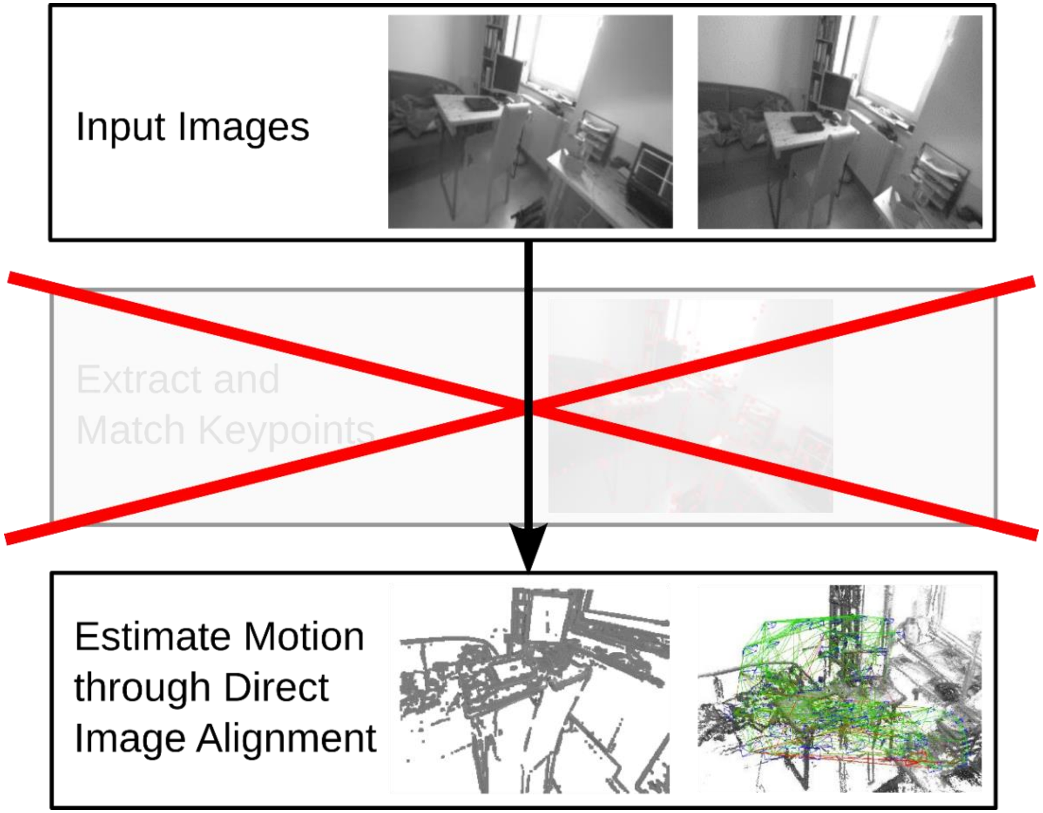
Direct Visual Odometry Pipeline

- Avoid manually designed keypoint detection and matching
- Instead: direct image alignment

$$E(\boldsymbol{\xi}) = \int_{\mathbf{y}\in\Omega} \left| I_1(\mathbf{y}) - I_2\big(\omega(\mathbf{y},\boldsymbol{\xi})\big) \right| d\mathbf{y}$$

$$E(\boldsymbol{\xi}) = \sum_i \left| I_1(\mathbf{y}_i) - I_2\big(\omega(\mathbf{y}_i,\boldsymbol{\xi})\big) \right|$$

- Warping requires depth
  - RGB-D
  - Fixed-baseline stereo
  - Temporal stereo, tracking and (local) mapping



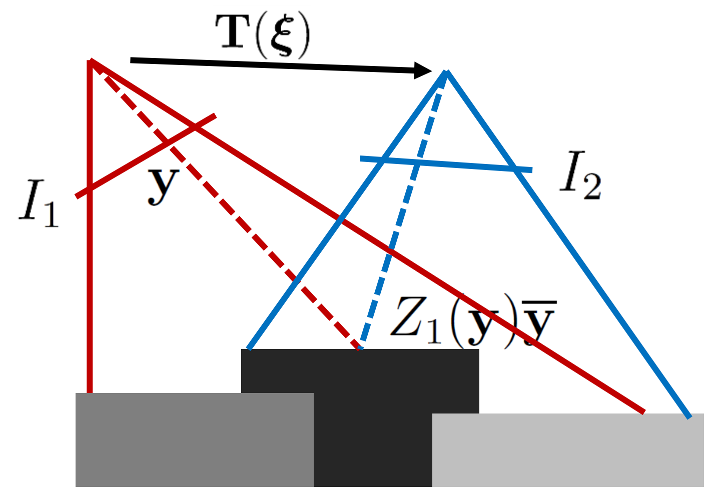
Probabilistic Direct Image Alignment

- Measurements are affected by noise

$$I_1(\mathbf{y}) = I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right) + \epsilon$$

- A convenient assumption is Gaussian noise

$$\epsilon \sim \mathcal{N}(0, \sigma_I^2)$$



- If we further assume that noise of pixel intensities is stochastically independent accross the image, we can formulate the a-posteriori probability

$$p(\boldsymbol{\xi} \mid I_1, I_2) \propto p(I_1 \mid \boldsymbol{\xi}, I_2)p(\boldsymbol{\xi})$$

$$\propto p(\boldsymbol{\xi}) \prod_{\mathbf{y}\in\Omega} \mathcal{N}\left(I_1(\mathbf{y}) - I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right); 0, \sigma_I^2\right)$$

Optimization Approach

- Optimize negative log-likelihood
  - Product of exponentials becomes a summation over quadratic terms
  - Normalizers are independent of the pose

$$E(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \Omega} \frac{r(\mathbf{y}, \boldsymbol{\xi})^2}{\sigma_I^2}$$, stacked residuals: $$E(\boldsymbol{\xi}) = \mathbf{r}(\boldsymbol{\xi})^\top \mathbf{W} \mathbf{r}(\boldsymbol{\xi})$$

$$r(\mathbf{y}, \boldsymbol{\xi}) = I_1(\mathbf{y}) - I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi}) Z_1(\mathbf{y}) \overline{\mathbf{y}}\right)\right)$$

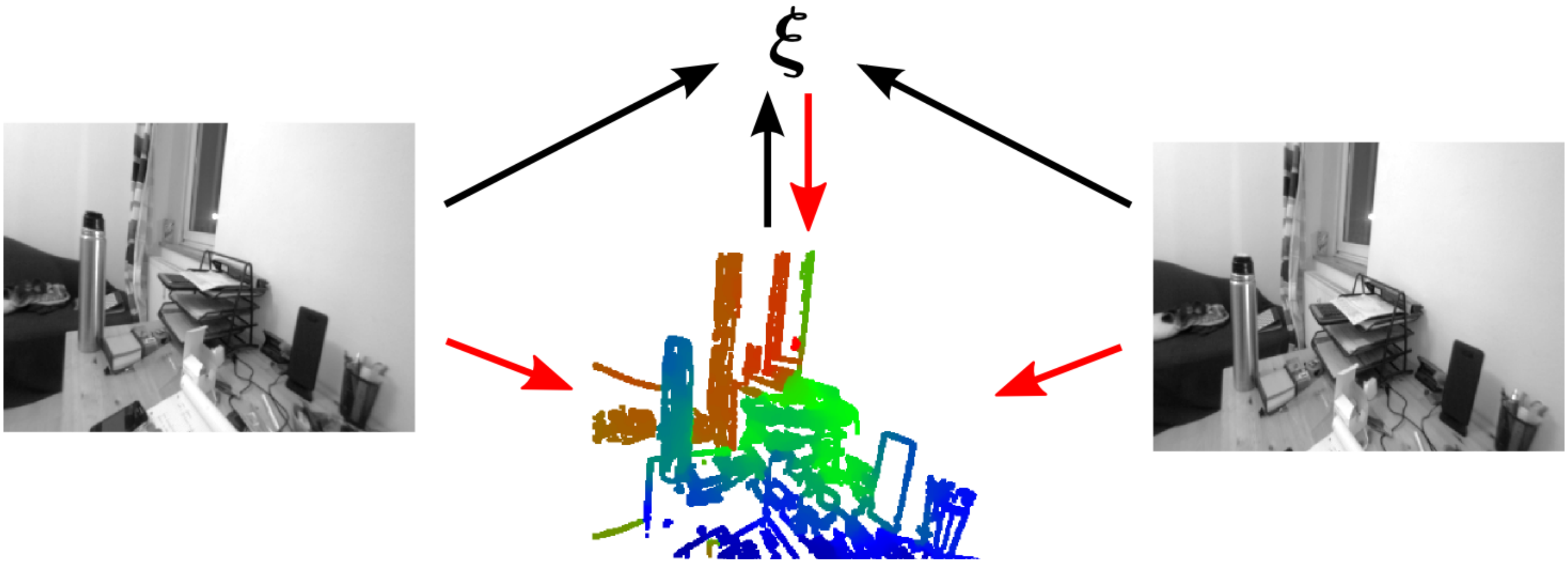
- Non-linear least squares problem can be efficiently optimized using standard second-order tools (Gauss-Newton, Levenberg-Marquardt)

Direct Visual Odometry

Direct RGB-D Odometry	Direct Monocular Odometry
Dense depth from sensor	Semi-dense depth estimated concurrently from short-baseline stereo comparisons and filtering
Only tracking of camera pose	Alternating, interdependent camera pose and depth map estimation
Tracking on keyframe	Tracking/depth estimation on keyframe
Metric scale from measured depth	No metric scale

Monocular Direct Visual Odometry

- Estimate motion and depth concurrently



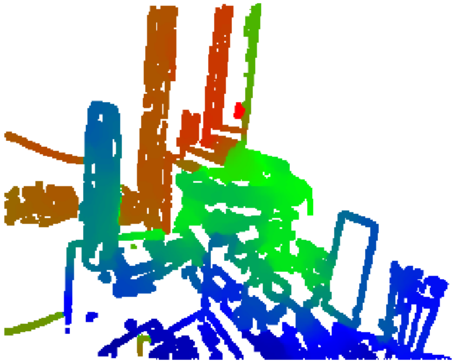
- Alternating optimization: **Tracking** and **Mapping**

Images from: Engel et al., ICCV 2013

Semi-Dense Mapping

- Estimate inverse depth and variance at high gradient pixels
- Correspondence search along epipolar line (5-pixel intensity SSD)





- Kalman-filtering of depth map:
  - Propagate depth map & variance from previous frame
  - Update depth map & variance with new depth observations

Images from: Engel et al., ICCV 2013

Visual-Inertial Fusion

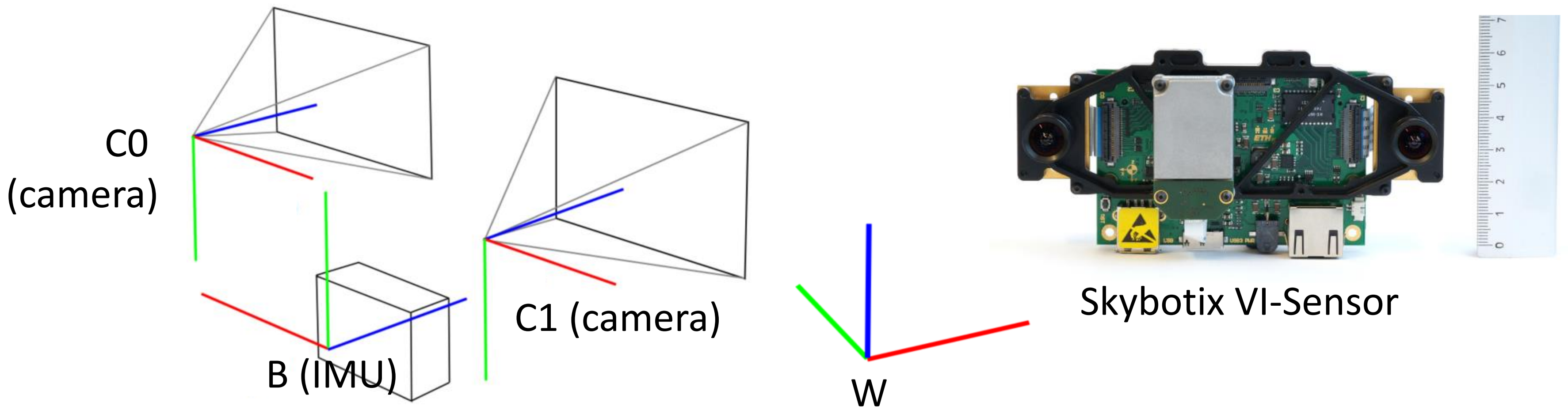
- Vision and IMU are complementary!

Visual sensing	Inertial sensing
+ Accurate at small to medium motion	- Large relative uncertainty for low acceleration/angular velocity
+ Rich information for other purposes	
- Limited output rate (~100Hz)	+ High output rate (~1000Hz)
- Scale ambiguity for monocular camera	+ Scale directly observable
- Lack of robustness for rapid motion, textureless areas, low illumination	+ Independent of environmental conditions

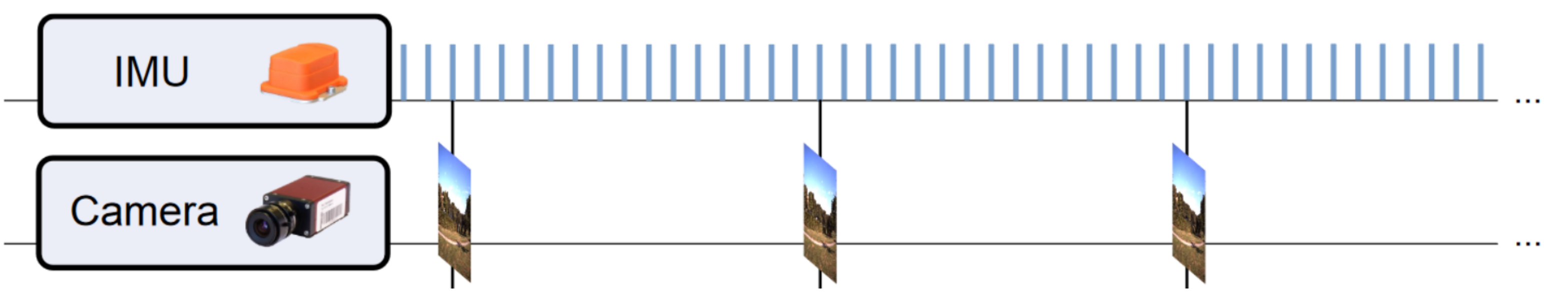
- Odometry using both sensor types is still prone to drift!

Camera-IMU System

- Extrinsic calibration between camera(s) and IMU frame

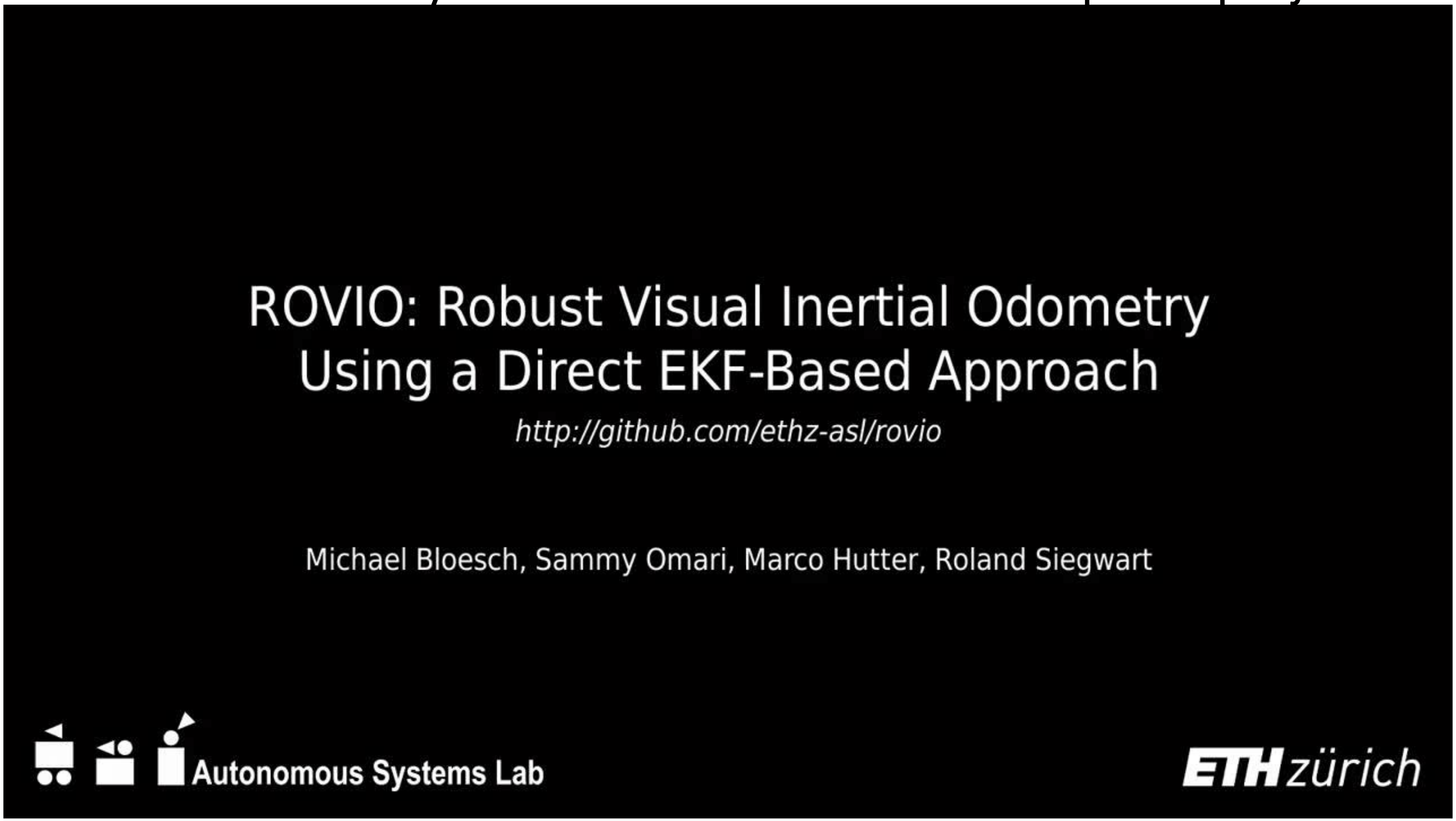


- Time synchronization



Tightly-Coupled Filter for Visual-Inertial Fusion

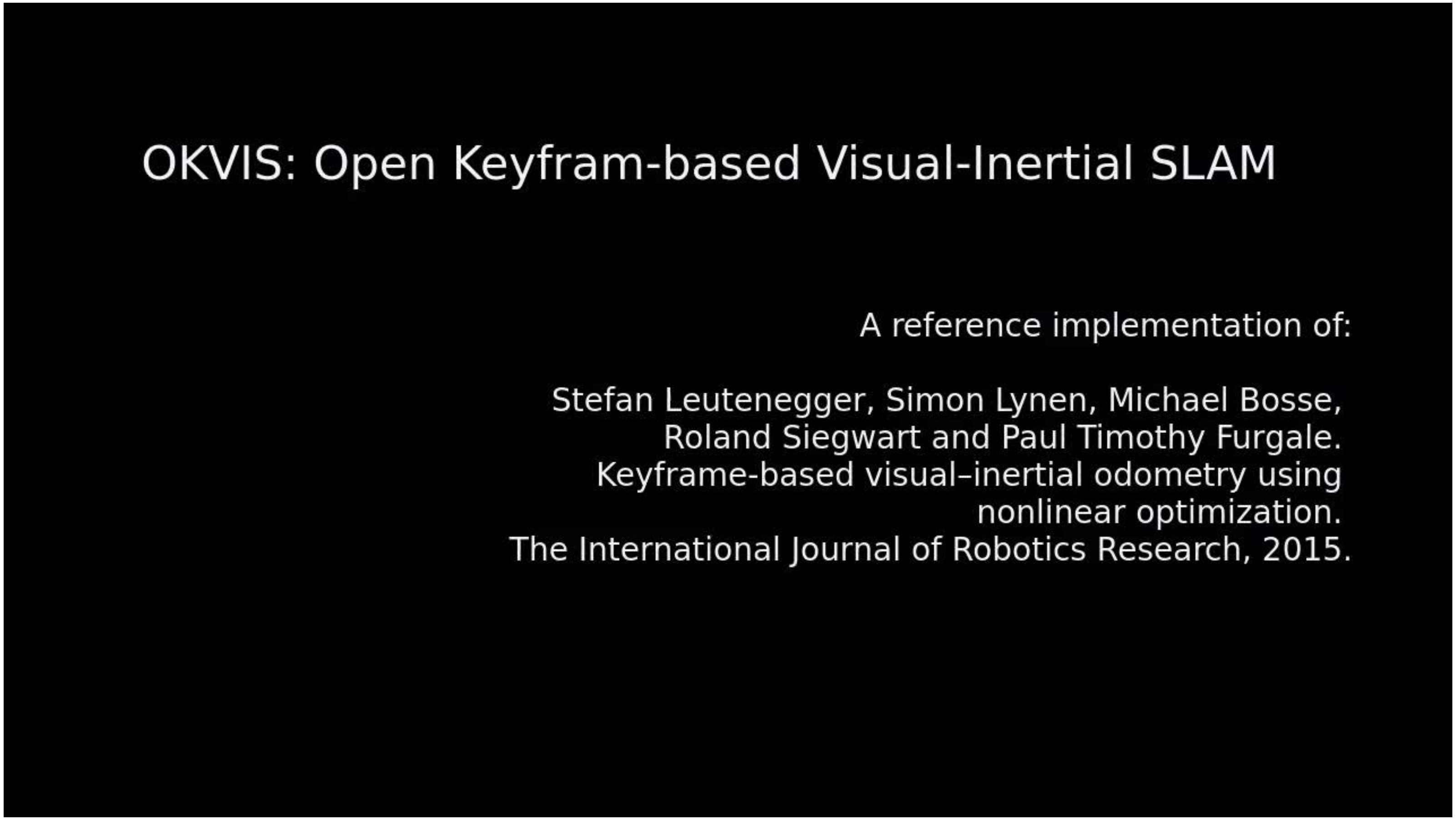
- Photoconsistency measurements of landmark patch projections



(Bloesh, Omari, Hutter, Siegwart, IROS 2015)
https://www.youtube.com/watch?v=ZMAISVy-6ao

Indirect Fixed-Lag Smoothing Example

- OKVIS: Keyframe-based indirect fixed-lag smoothing VIO

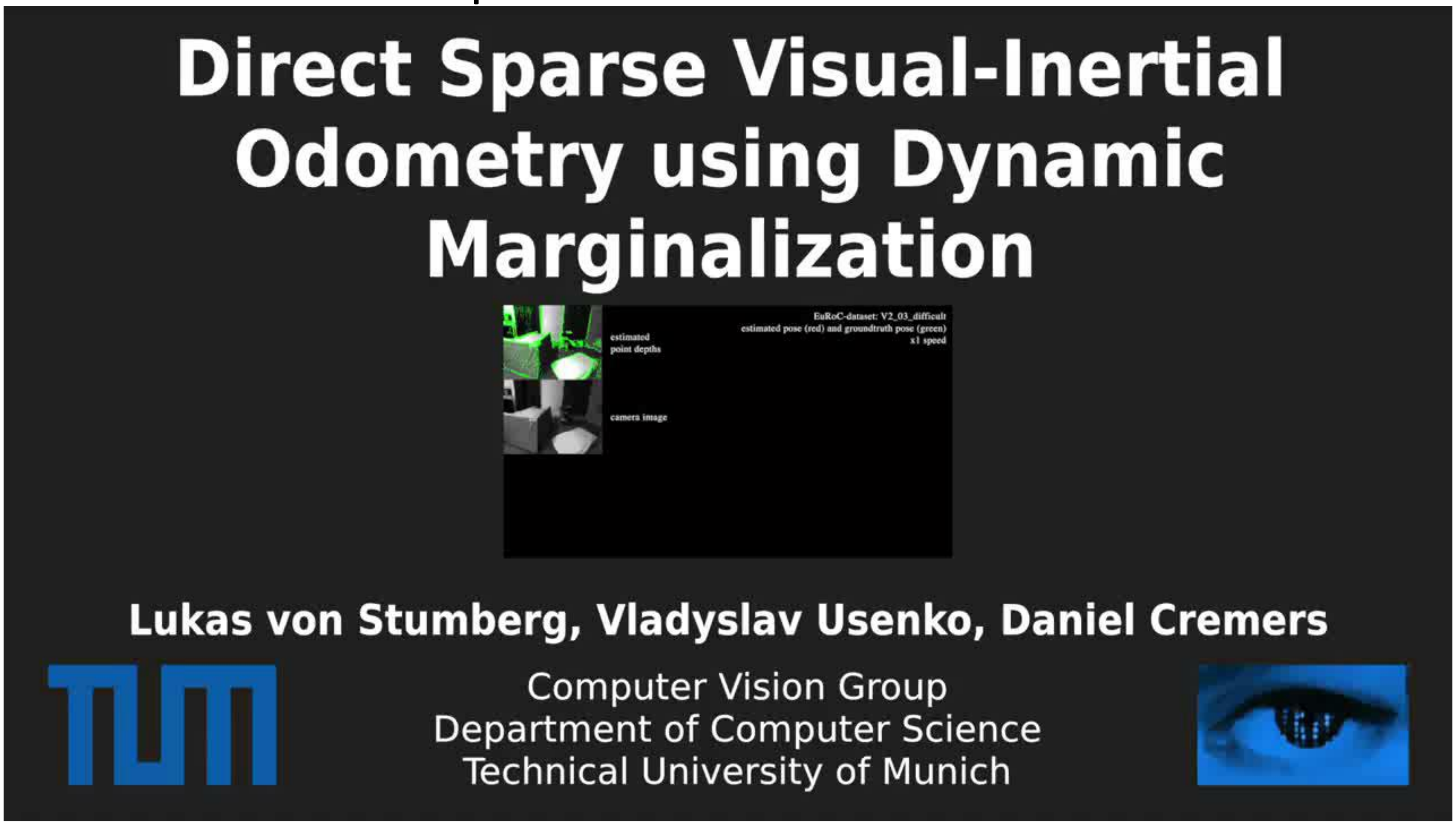


(Leutenegger, Lynen, Bosse, Siegwart, Furgale, IJRR 2015)

https://www.youtube.com/watch?v=TbKEPA2_-m4

Fixed Size Optimization Window Example

- Direct Fixed Size Optimization Window VIO



(von Stumberg, Usenko, Cremers, ICRA 2018)

https://www.youtube.com/watch?v=GoqnXDS7jbA

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

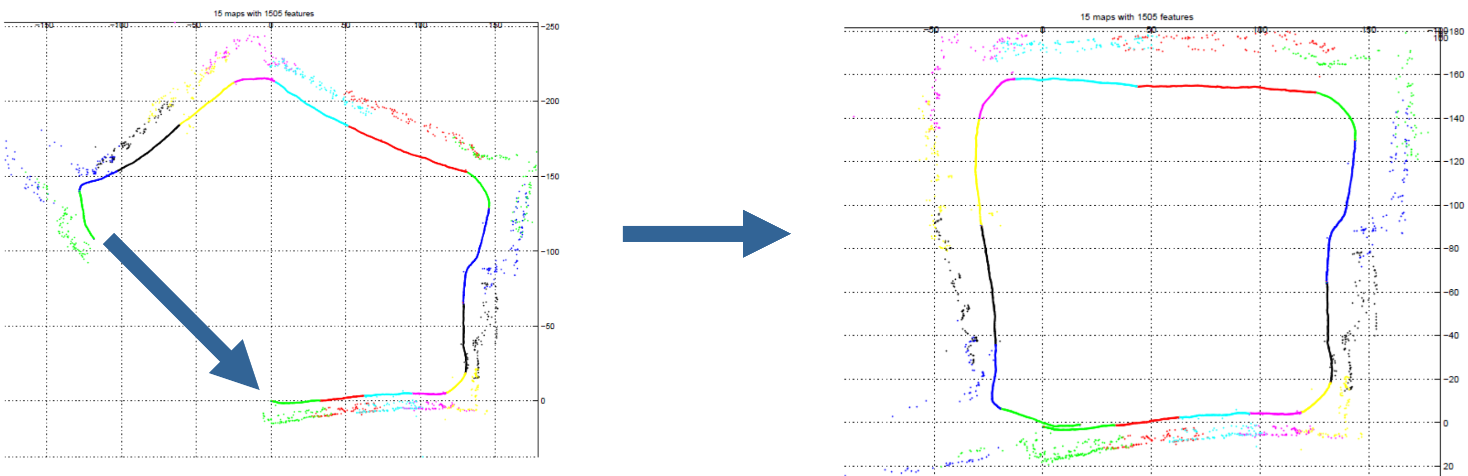
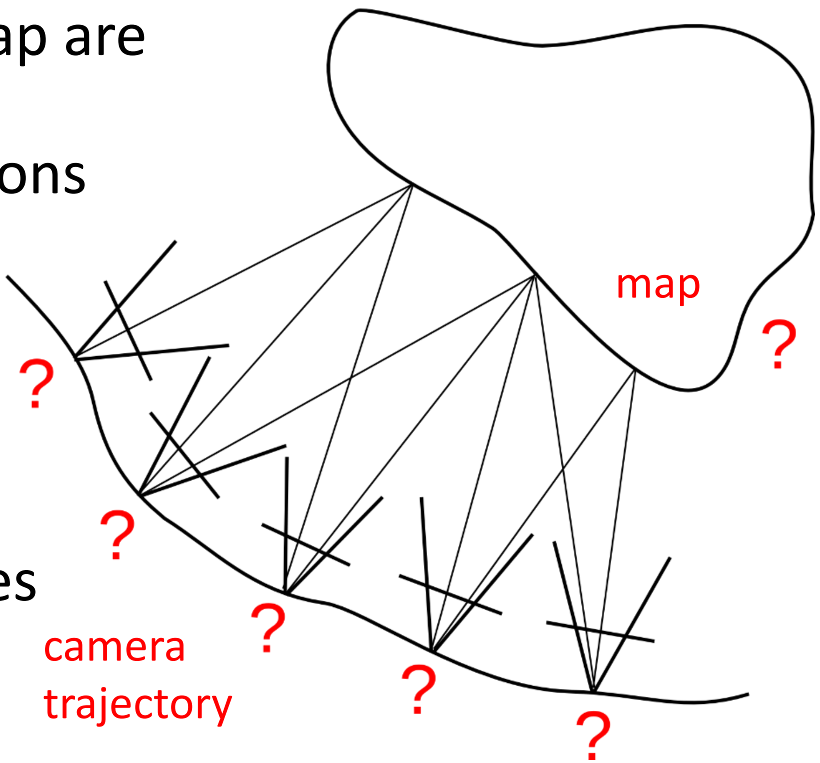


Image credit: Clemente et al., RSS 2007

Why is SLAM difficult?

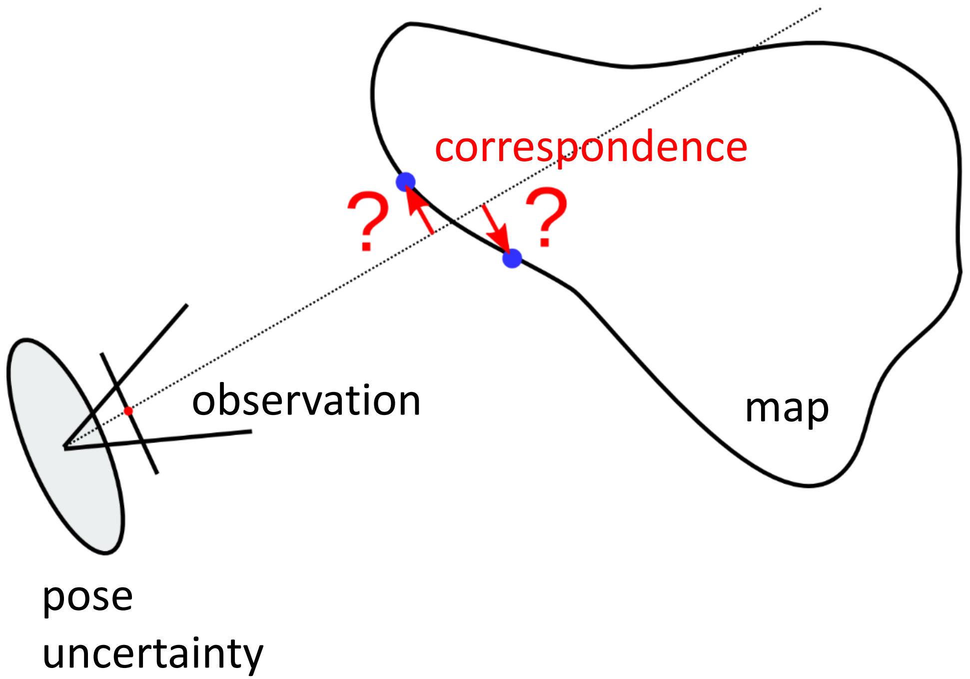
- Chicken-or-egg problem
  - Camera trajectory and map are unknown and need to be estimated from observations
  - Accurate localization requires an accurate map
  - Accurate mapping requires accurate localization



- How can we solve this problem efficiently and robustly?

Why is SLAM difficult?

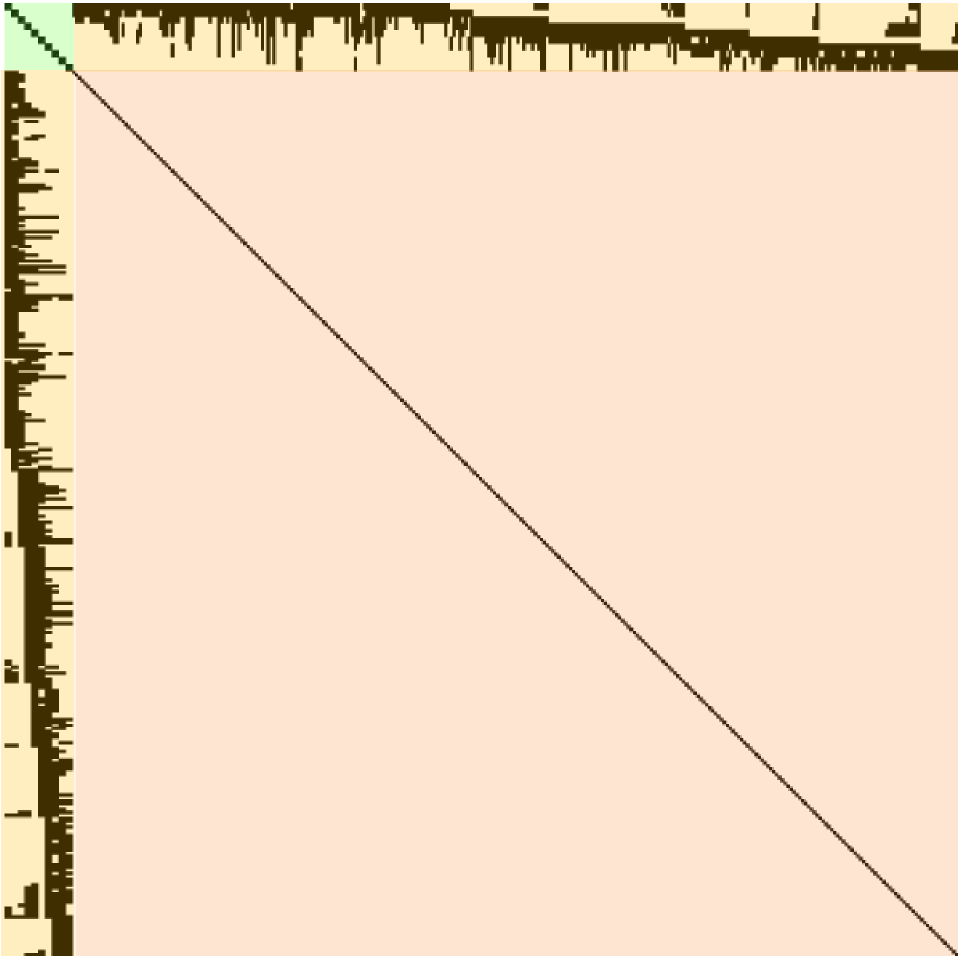
- Correspondences between observations and the map are unknown
- Wrong correspondences can lead to divergence of trajectory/map estimates
- Important to model uncertainties of observations and estimates in a probabilistic formulation of the SLAM problem



Example Hessian of a BA Problem

Pose dimensions

(10 poses)



Landmark dimensions

(982 landmarks)

Image source: Manolis Lourakis (CC BY 3.0)

Exploiting the Sparse Structure

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

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

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

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

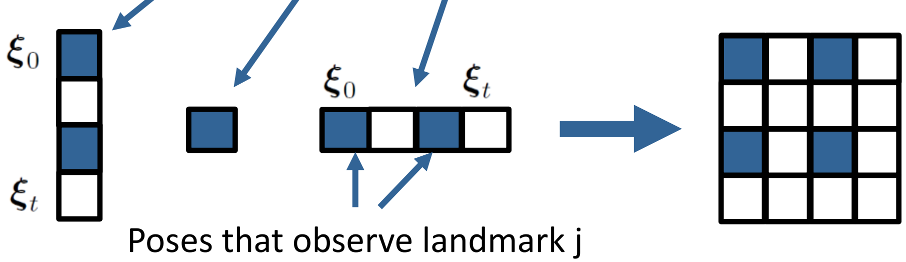
- Is this any better?

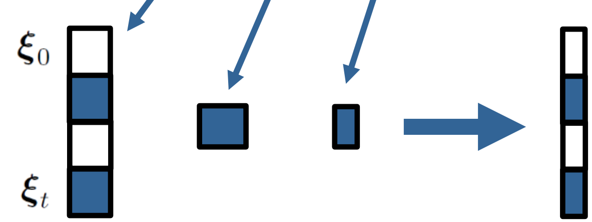
Exploiting the Sparse Structure

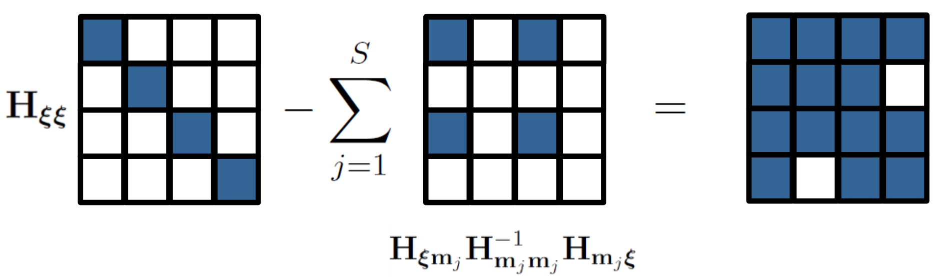
- What is the structure of the two sub-problems ?
- Poses: $\Delta \mathbf{x}_\xi = -\left(\mathbf{H}_{\xi\xi} - \mathbf{H}_{\xi \mathbf{m}}\mathbf{H}_{\mathbf{mm}}^{-1}\mathbf{H}_{\mathbf{m}\xi}\right)^{-1}\left(\mathbf{b}_\xi - \mathbf{H}_{\xi \mathbf{m}}\mathbf{H}_{\mathbf{mm}}^{-1}\mathbf{b}_\mathbf{m}\right)$

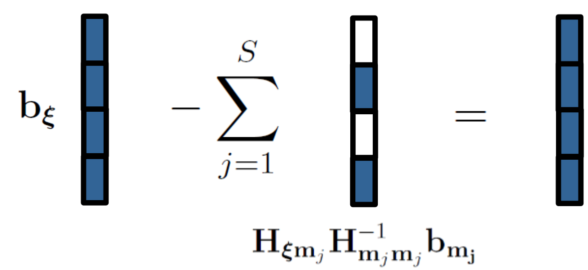
$$\mathbf{H}_{\xi\xi} - \sum_{j=1}^{S} \mathbf{H}_{\xi \mathbf{m}_j}\mathbf{H}_{\mathbf{m}_j\mathbf{m}_j}^{-1}\mathbf{H}_{\mathbf{m}_j\xi}$$

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

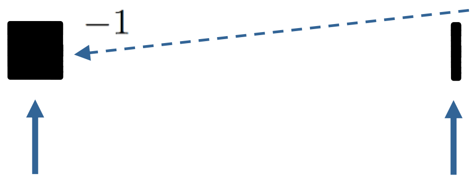








Exploiting the Sparse Structure



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

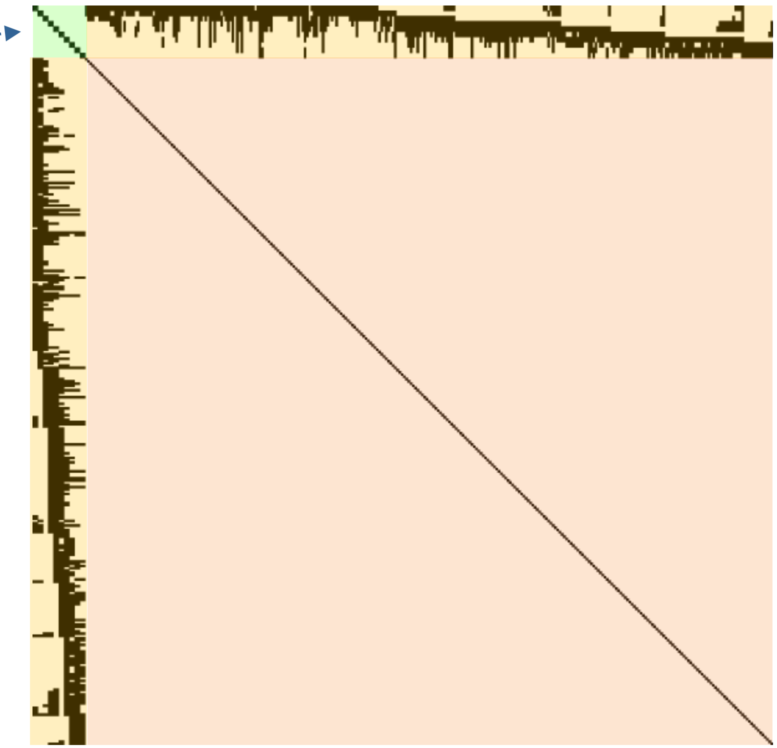
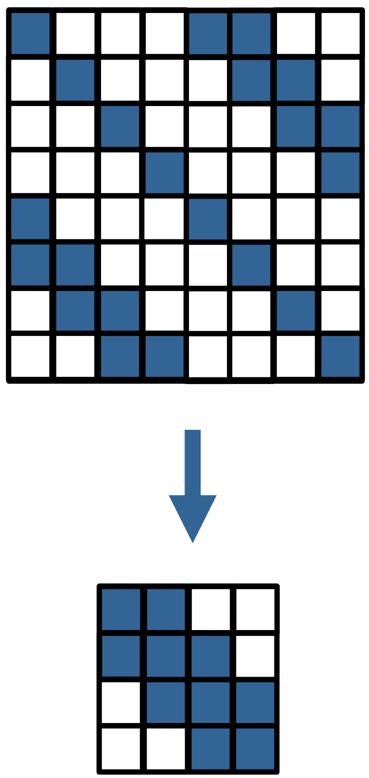
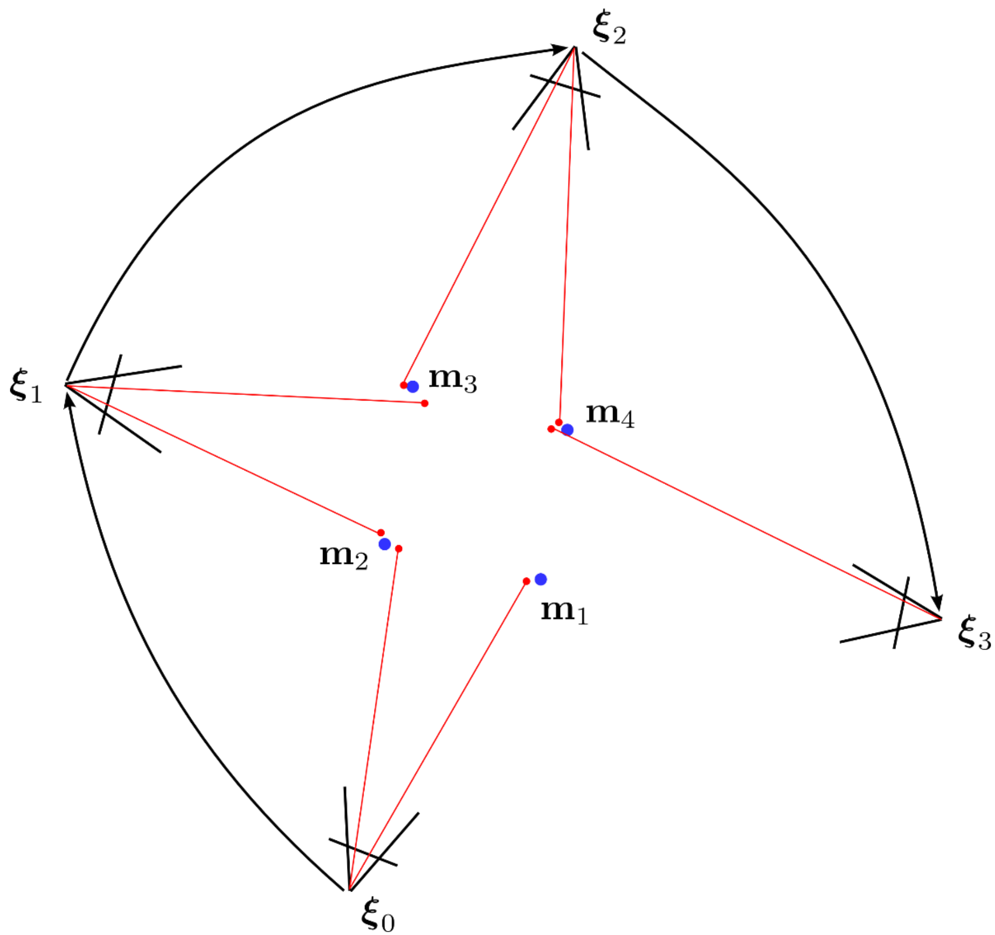


Image source: Manolis Lourakis (CC BY 3.0)

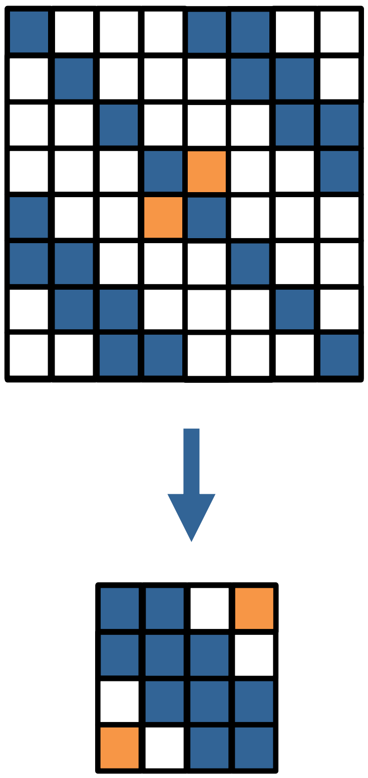
Effect of Loop-Closures on the Hessian



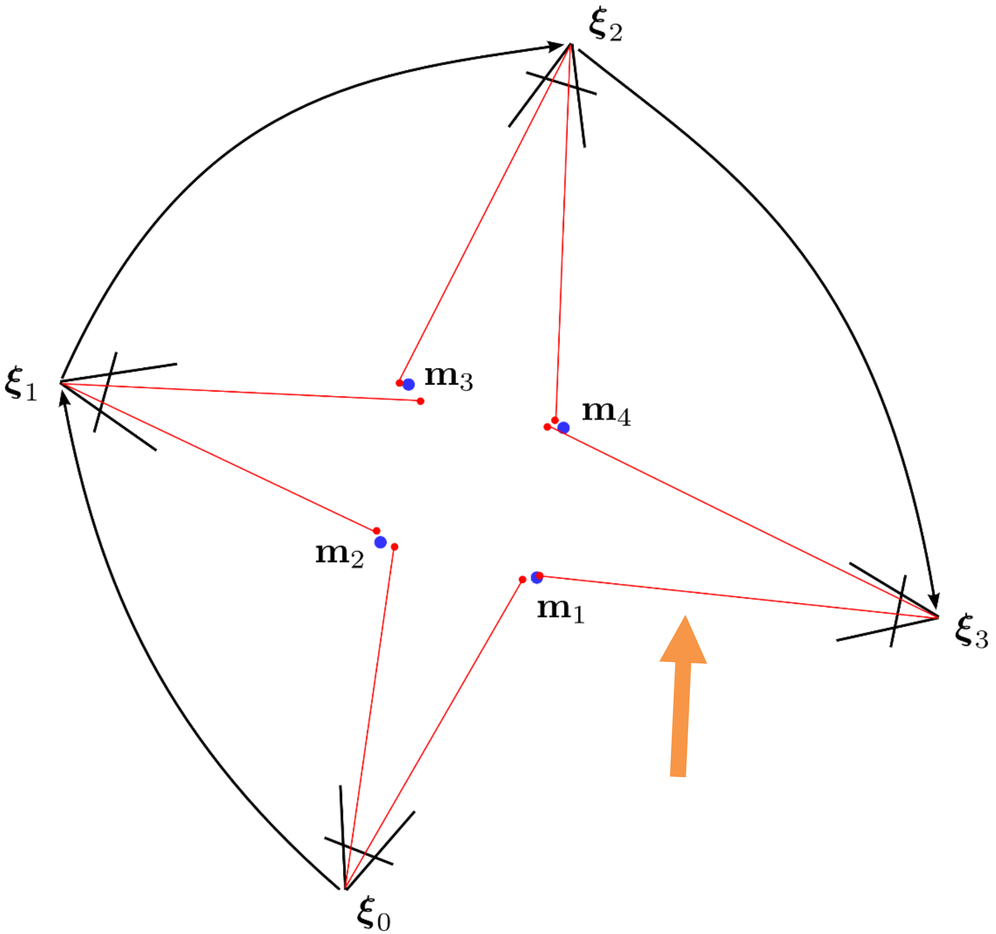
Band matrix



Effect of Loop-Closures on the Hessian



Not band matrix: costlier to solve



Loop Closing by Place Recognition

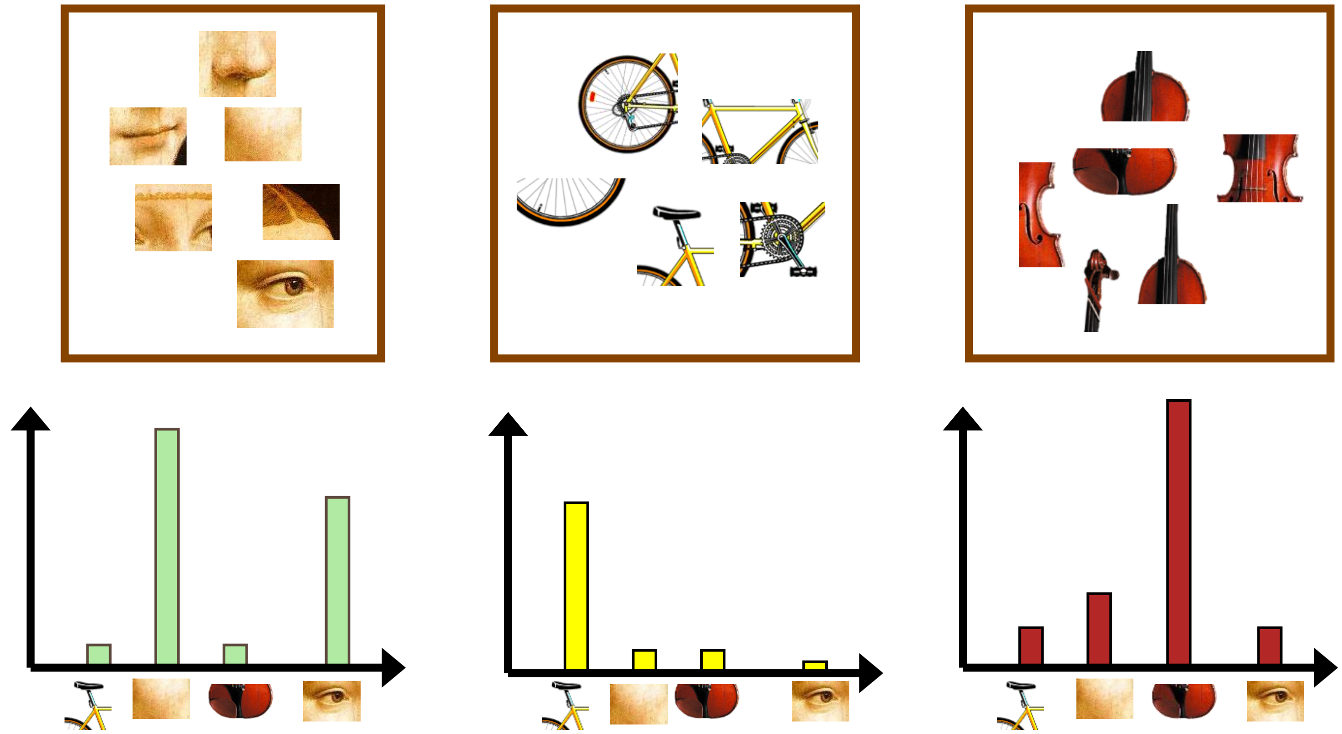
- Idea: use image retrieval techniques

- Popular approach for place recognition is to use bag-of-visual-words based image retrieval in conjunction with geometric verification (f.e. 8-point with RANSAC)

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

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

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).